Multistep Methods for Stiff Initial Value Problems

Dana Petcu

October 25, 1994

Contents

Part I

PROBLEM DEFINITION AND CLASSICAL SOLUTIONS

Chapter 1

STIFF? WHAT IS IT?

1.1 The initial value problem

A stiff problem is a particular case of an initial value problem. We define the CAUCHY's problem or *initial value problem* a system with an arbitrary number of ordinary differential equations of first order endowed with a given starting value:

$$\begin{cases} y'(t) = f(t, y(t)), & a \leq t \leq b, \\ y(a) = y_a. \end{cases} \tag{1.1}$$

In this case, there are N equations and $y : [a, b] \to \mathbf{R}^N$, $f : [a, b] \times \mathbf{R}^N \to \mathbf{R}^N$, $y_a \in \mathbf{R}^N$.

The step by step numerical algorithm builds an approximate solution by an interative procedure starting from $t = a$. Note $y(t_n)$ the exact solution at the point t_n of the associated interval division

$$a = t_0 < \cdots < t_n < \cdots < t_M = b,$$

and y_n, the approximate value computed with the numerical method.

For the unicity of the exact solution we put the following conditions:

- the system function f is continous on

  $$D = \{(t, y) \,|\, t \in [a, b],\ \|y(t) - y\| \leq d\},$$

  where $d \in \mathbf{R}_+$;

- f satisfies the LIPSCHITZ's inequality

  $$\|f(t, u) - f(t, v)\| \leq L\|u - v\|, \quad a \leq t \leq b, \quad \forall (t, u), (t, v) \in D \tag{1.2}$$

For almost all differential equation systems, the value $L(b-a)$ is of hundreds order. For such systems the classical step by step methods, like RUNGE-KUTTA's processes or ADAMS's formulae, are satisfactorily (in the sense of a small error).

Unfortunately, there are some exceptions, for instance, the case when the function variation is very strong. When the value $L(b - a)$ exceeds the thousands order, a

variety of restrictions are imposed to the classical methods, especially on the stepsize, so there are useless. Such a system we classified to be of stiff kind one.

Remark. Applying (1.2), we deduce the existence of a value $\Delta > 0$ so that the inequality

$$\|u(t+\Delta) - v(t+\Delta)\| \leq \|u(t) - v(t)\|, \quad \forall t \in [a,b], \tag{1.3}$$

holds for two arbitrary exact solutions of the differential system, u and v, which are starting from different initial values. The value Δ depends on L. If the value L is large, the value Δ is very small. The inequality (1.3) is important for the numerical integration because it tell us that the approximate solution can follows an other exact solution which start from an initial value which is closed to that of the exact solution and the approximation error (the difference between the approximate solution and the exact solution) can be controlled.

A more general condition is that of the *dissipativity* of the system function:

$$< f(t,u) - f(t,v), u - v > \leq \mu \|u - v\|^2, \tag{1.4}$$

where μ is the *constant of dissipativity.* When $y, f(t,y) \in \mathbf{C}^N$, the dissipativity condition can be reformulated:

$$\mathrm{Re} < f(t,u) - f(t,v), u - v > \leq l \|u - v\|^2. \tag{1.5}$$

A LIPSCHITZ continous function satisfies the dissipativity condition, but the reverse implication is not valid. The value μ can be lowest than the value L, also a negative one.

Remark. The dissipativity condition confirms the existence of a value Δ for which

$$\|u(t+\Delta) - v(t+\Delta)\| \leq e^{\mu\Delta} \|u(t) - v(t)\|, \quad \forall t \in [a,b]. \tag{1.6}$$

For $\mu \leq 0$, the difference between two arbitrary solution is a decreasing function.

1.2 The stability of the exact solutions

Let $y' = \lambda y,\ \lambda \in R$. The solutions family of this equation, can be represented by $\{y \,|\, y(t) = ce^{\lambda t},\ c \in \mathbf{R}\}$. Note that, when $\lambda > 0$, for some large t values, the difference between two arbitrary solutions increases exponentially, even if the difference between the initial values is very small. In such a case, applying an approximate method, the results are unforeseeables. If $\lambda < 0$, the difference between two arbitrary exact solutions decreases when $t \to \infty$. These remarks can be extended for the case

$$y' = \lambda y,\ y \in C \tag{1.7}$$

(discussion on $\mathrm{Re}\,\lambda > 0$ or $\mathrm{Re}\,\lambda < 0$). An equation of (1.7) form is referred to as the *scalar test equation.*

The scalar test equation is used in the numerical stability study of the step by step methods, especially for the multistep formulae.

Let $f : [t_0, \infty) \times \mathbf{R}^N \to \mathbf{R}^N$, $y(\cdot, t_0, y_0)$ the exact solution of the CAUCHY's problem with $y(t_0) = y_0$, and φ an arbitrary solution for the differential system.

Definition 1.1. The solution φ is *stable* in LIAPUNOV's sense if, for any $\varepsilon > 0$, there is a value $\delta(\varepsilon, t_0)$ such that

$$\|\varphi(t) - y(t, t_0, y_0)\| < \varepsilon, \quad \forall t \in [t_0, \infty), \tag{1.8}$$

holds for any y_0 satisfying $\|y_0 - \varphi(t_0)\| \leq \delta(\varepsilon, t_0)$. The solution φ is *asympotically stable* if, for any $\varepsilon > 0$, there is a value $\delta(t_0)$ such that (1.8) holds for any y_0 satisfying $\|y_0 - \varphi(t_0)\| < \delta(t_0)$.

In the particular case of a linear system, $y' = Ay$, if the solution $y = 0$ is stable (asymptoticallly stable), all the system solutions are stable (asymptotically stable). The solution $y = 0$ is asymptotically stable if all the eigenvalues of the matrix A have some negative real parts, that means that the characteristic polynomial of A is an HURWITZ polynomial.

The stability of the exact solutions is out of the present study subject.

The following supposition is esentially in our study.

The exact solutions of the numerical solved problems with initial values are asymptotically stable

By this supposition we eliminate the possibility of a numerical instability due to the instability of the exact solutions.

1.3 The stability of the numerical solutions

If the error propagation mechanism is under control, the numerical method is stable. In mathematical terms, this fact can be expressed in the following form.

Definition 1.2. A method (formula, process, scheme) is *stable* if, for each differential equation with all solutions asymptotically stable ones, there are two values, K and h_0, so that

$$\|y_n - \overline{y}_n\| \leq K\|y_0 - \overline{y}_0\|, \quad \forall h : \ 0 < h < h_0, \tag{1.9}$$

holds, where y_n and $\overline{y}_n$ are the values of the approximate solutions in the n^{th} point of the interval division, computed with the stepsize h and starting from the value y_0, respectively $\overline{y}_0$. The method is *asymptotically stable* if, for a given stepsize, the perturbations in the numerical solution do not grow from a step to another, i.e.

$$\|y_n - \overline{y}_n\| \leq \|y_{n-1} - \overline{y}_{n-1}\|. \tag{1.10}$$

Remark. The numerical stability in the sense of definition 1.2, tell us that a small perturbation produced to one step of integration determines only a small perturbation in the numerical solution for all future points, when we use a small stepsize.

1.4 Order and consistency

Applying the numerical method, we find out some approximate values y_n of the exact values $y(t_n)$. Ideally, $y_n = y(t_n)$, i.e. the numerical algorithm is exact. Unfortunately, this request can be satisfied only in the case when we use a multistep formula to integrate an equation with a polynomial solution. Generally, the difference between the approximate solution and the exact solution must vanish when the division points number becomes infinitely. That means that the approximation can be improved growing the number of the division points, i.e. increasing the computational effort. The *consistency* express this request in the theoretical case when the numerical algorithm can be applied without truncation errors.

The *method order* is the convergence order of the approximate values set at $h \to 0$ $(n \to \infty)$.

Example. Let the onestep method

$$y_{n+1} = y_n + h\Phi(t_n, y_n, h). \tag{1.11}$$

Note

$$z_{n+1} := y(t_n) + h_{n+1}\Phi\left(t_n, y(t_n), h_{n+1}\right), \quad TE = y(t_{n+1}) - z_{n+1}.$$

The value TE is the *local discretization error*. The *global error* can be expressed by the difference

$$y(t_{n+1}) - y_{n+1} = TE + (z_{n+1} - y_{n+1})$$

If

$$TE = Ch^{p+1}\varphi(t_n, y(t_n)) + O\left(h^{p+2}\right),$$

where φ depends on the system function (for almost all multistep formulae, is a p order derivative of the system function), then the algebraic condition of consistency is $p \geq 1$. The p value is the *accuracy order* of the numerical method.

The notion of *accuraccy* supposes the consistency condition and a small error constant C, smallest than the stepsize h.

1.5 Behaviour of classical methods for the stiff case

We study a suggestive example. Let

$$y' = Ay, \qquad t \in [0, T], \quad y(0) = y_0$$

and a division of the integration interval with the constant stepsize $h > 0$ and the points $t_n = nh$, $n \geq 0$. The exact solution y satisfies the reccurence relation $y(t_{n+1}) = e^{Ah}y(t_n)$. The most simple numerical procedure, for finding an approximate value y_n of the exact value $y(t_n)$, is the EULER's *explicit rule*. For the given system, the iterative process is

$$y_{n+1} = y_n + hAy_n, \qquad n \geq 0$$

Set $S(z) = e^z$, $R(z) = 1 + z$. Then the reccursive equations of the solutions are the followings:

$$y(t_{n+1}) = S(hA)y(t_n), \qquad y_{n+1} = R(hA)y_n.$$

The global error at the point t_n is the value $e_n = y(t_n) - y_n$. The reccursive equation is $e_{n+1} = R(hA)e_n + T(hA)y_n$, where $T = S - R$ is the *truncation function*. The solution of the reccursive equation starting from $e_0 = 0$ is $e_{n+1} = \sum_{j=0}^{n} R^j(hA)T(hA)y_{n-j}$. Thus, the global error can be approximated by

$$\|e_n\| \leq n \max_{0 \leq j \leq n-1} \|R(hA)\|^j \max_{0 \leq j \leq n-1} \|T(hA)y_j\|.$$

If the numerical solution is *stable*, i.e. $\|R(hA)\| \leq 1$, and is of p *order of consistency*, i.e. $\|T(hA)y\| = \mathcal{O}(h^{p+1})\|y\|$, then an approximation of the global error is $\|e_n\| = \mathcal{O}(h^p)$ and the method converges, i.e. $e_n \to 0$, when $n \to \infty$. For the EULER's explicit rule $p = 1$, since

$$\|T(hA)\| = \mathcal{O}(|\lambda_{max}|h^2), \quad |\lambda_{max}| = \max_{1 \leq j \leq N} |\lambda_j|,$$

where λ_j are the eigenvalues of the system matrix, A. Thus we can explain the use of the consistency order as a measurement of the convergence order of the method.

The stability condition $\|R(hA)\| \leq 1$ means that

$$|1 + h\lambda_j| \leq 1, \qquad j = 1, \ldots, N.$$

The set

$$A = \{z \,|\, |R(z)\| \leq 1\}.$$

is the *absolute stability domain*. For the EULER's explicit rule, $R(z) = 1 + z$, and the stability domain is the set of complex values which are inside to the unitary circle $\{z \in \mathbf{C} \,|\, |1 + z| = 1\}$.

For usual differential equations, $|\lambda_{max}| = \max_{1 \leq i \leq N} |\lambda_i|$ is not a very large value and the restriction imposed by the stability condition, $z_i = h\lambda_i \in A$, $i = 1, \ldots, N$, is satisfied for some reasonable stepsizes. A classical example of stiff system is the one for which $|\lambda_{max}|$ is very large. In this case, neither the stability condition, nor the error boundness can not be satisfied without the strong restriction of the stepsize, in fact introducing a large computational overhead. The term "large" is relative to the value $1/T$. Indeed, also an equation with a $|\lambda_{max}|$ small value can be classified as a stiff one if the requested solution is defined on a very large integration interval.

Remark. Let the CAUCHY's problem associated to the nonlinear equation $y' = f(t, y)$ with φ, a numerical solution. In a $\varphi(t)$ neighbourhood we can approximate the derivative

$$y'(t) \approx f(t, \varphi(t)) + J(t)(y(t) - \varphi(t)).$$

where $J(t)$ is the JACOBIAN's matrix $(\partial f(t, y)/\partial y)(t, \varphi(t))$. Then the global error $e(t) = y(t) - \varphi(t)$ can be approximated by

$$e'(t) \approx J(t)e(t).$$

Thus, the error evolution depends on the eigenvalues of the matrix $J(t)$.

A *numerical method for solving stiff problems* is a method for which the stability condition does not ask that $|h\lambda_j|$ remains small also when the corresponding component in the exact solution is negligible. Practically, there are such methods. For example, the EULER's *implicit rule*, with the reccursive relation

$$y_{n+1} = y_n + hf_{n+1}, \tag{1.12}$$

where $f_{n+1} = f(t_{n+1}, y_{n+1})$, has the requested property. The stability domain is the set of the complex points lying outside the unitary circle $\{z \in \mathbf{C} \,|\, |1 - z| = 1\}$.

The stepsize restriction is only one characteristic of the stiff phenomenon.

The numerical methods based on polynomial interpolation are succesfully in the integration of a stiff system only if they are implicit, and, consequently, suppose some procedures for solving the nonlinear difference equations. Solving implicit nonlinear equations associated to stiff problem, the use of the method of simple iterations is out of question. For example, for the formula (1.12), the method of simple iterations concerns the iterative process

$$y_{n+1}^{(m+1)} = y_n + hf\left(t_{n+1}, y_{n+1}^{(m)}\right), \quad m \geq 0$$

where $y_{n+1}^{(0)}$ is given by an explicit method. The convergence condition of this iterativ process is

$$h\rho\left(\frac{\partial f}{\partial y}(t_{n+1}, y_{n+1})\right) < 1,$$

which can not be satisfied for a reasonable value h relative to the integration interval length. The most popular method for solving the implicit equations, in the stiff case, is the NEWTON's iterative process. The JACOBIAN's matrix must be computed to each iterative step, which is very expensive in the sense of the computational effort.

1.6 Definition

The essence of the stiff phenomenon consists in the fact that the exact solution includes some components with a very fast decreasement that can be very hard to be followed by the numerical solution given by a step by step iterative process.

We mention some pragmatic definitions:

- The stiff equations are the equations for which some implicit methods work better than the explicit ones.
- A problem is stiff in a given integration interval if, for a given numerical code, the stepsize must be very strongly reduced.
- The stiff differential equations are wrong conditioned in the computational sense.
- An ordinary differential equation system is stiff in a given integration interval $[0, T]$ if, in the exact solution, there is at least one component with a very large variation relative to the value $1/T$.

Example 1. Let the scalar equation

$$y'(t) = f(t, y) = \lambda y(t) + F'(t) - \lambda F(t), \quad t \geq 0, \quad \lambda \ll 0.$$

The exact solution of the associated CAUCHY's problem is

$$y(t) = F(t) + e^{\lambda t}[y_0 - F(0)],$$

We notice that, after a short time, *the transitory component* $e^{\lambda t}[y_0 - F(0)]$, also referred to as *the stiff component*, has no more a signifiant influence on the solution value. There is a subinterval in which the slow varying component, $F(t)$, also referred to as *the untransitory component* or *the smooth component*, is prevalent in the exact solution. This subinterval covers almost all the integration interval. Applying the EULER's explicit rule with stepsize h to this problem, we get

$$y_{n+1} = y_n + hf(t_n, y_n) = (1 + h\lambda)[y_n - F(t_n)] + F(t_n) + hF'(t_n).$$

If the numerical model is correct, the difference $y_n - F(t_n)$ must decrease: this is true iff $-2 < h\lambda < 0$. Thus, *the stepsize is retricted by some numerical stability condition.*

In the transitory stage a problem can not be classified to be stiff one, only when the stiff component is negligible. An equation can not be referred to as stiff one (only the CAUCHY's problem associated to it), since the term is relative to the initial value, the integration interval and the error tolerance.

Example 2. Some components with different dials of variation can be meet especially in the case of the linear systems of the following form

$$y'(t) = Ay(t) + F(t), \tag{1.13}$$

where A is a constant matrix with different dials values of the eigenvalues λ_i. The problem (1.13) is stiff if (FATUNLA[43])

(i) $\exists i$: $\quad \mathrm{Re}\,(\lambda_i) \ll 0$;

(ii) $\exists j$: $\quad |\lambda_j| \ll |\lambda_i|$;

(iii) $\nexists k$: $\quad \mathrm{Re}\,(\lambda_k) \gg 0$;

(iv) $\nexists m$: $\quad |\mathrm{Im}\,(\lambda_m)| \gg 0$, only if $\mathrm{Re}\,(\lambda_m) \ll 0$.

We define the *stiff ratio* of the system (1.13) by

$$S(A) = \max_{j=1,\ldots,N} |\mathrm{Re}\,(\lambda_j)| / \min_{j=1,\ldots,N} |\mathrm{Re}\,(\lambda_j)|.$$

The system (1.13) is stiff if $\mathrm{Re}\,(\lambda_j) < 0$, $j = 1, \ldots, N$, and $S(A) \gg 1$ (CASH[25]).

Example 3. An important class of stiff problems with very large dimensions is derived from some partial differential equations. For example, let the problem posed in [53]

$$u_t = u_{xx}, \ t > 0, \ 0 \leq x \leq 1, \ u(t, 0) = u(t, 1) = 0, \ t > 0, \ u(0, x) = \varphi(x)$$

and $u_i(t)$ the approximate values of the exact values $u(t, x_i)$, produced by replacing u_{xx} with

$$\frac{\delta^2 u(t,x_i)}{h^2} = \frac{u(t,x_i+h) - 2u(t,x_i) + u(t,x_i-h)}{h^2}, \quad x_i = x_{i-1} + h,$$

where h is the stepsize on x axis, $h = 1/m,\ m \in \mathbf{N}$. Then $u_i(t), i = 1, \ldots, m-1$ are the components of the exact solution of the CAUCHY's linear problem

$$y'(t) = Ay(t), \quad t > 0, \quad A = m^2 \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}.$$

The eigenvalues λ_i of the matrix A,

$$\lambda_i = m^2 \left[-2 + 2\cos\left(\frac{i\pi}{m}\right) \right], \qquad i = 1, \ldots, m-1,$$

are uniformly distributed in the interval $(-4m^2, 0)$. When m increases (the division is more dense) the stiff ratio also increases.

Example 4. The singular perturbation problems are particular cases of stiff problems. The generic form of a such problem is [53]

$$\begin{cases} x' = f(t,x,y,\epsilon), \\ \epsilon y' = g(t,x,y,\epsilon,) \end{cases} \qquad \begin{array}{c} x(0) = \varepsilon, \\ y(0) = \eta, \\ g(t,x,y,0) = 0. \end{array}$$

In fact this system is a stiff one. For example, let $f = g = x + y$. Then the eigenvalues of the system are $\epsilon^{-1} + \mathcal{O}(1)$ and $-1 + \mathcal{O}(\epsilon)$. When ϵ decreases, the stiff ratio is increasing.

Such systems can be solved by some *pseudo-stationary methods*. We consider the reduced problem

$$\begin{cases} x' = f(t,x,y,0), \\ 0 = g(t,x,y,0), \\ x(0) = \varepsilon. \end{cases}$$

Unfortunately, there are some systems which can not be simplified in this manner since the presence of the control value ϵ determines the evolution of the exact solution.

Example 5. If the system presents a variation on t, like

$$y'(t) = A(t)y(t),$$

the affiliation to the stiff class is not easy to be establish. The problem is only *local* a stiff one. The variation of the eigenvalues of the JACOBIAN's matrix can not inform us about the perturbation propagation in the exact solution. In practice, a system is classified to be stiff one if, at least at a moment (at one t from the integration interval), the associated initial value problem is a locally stiff problem.

Table 1.1: Subclasses of stiff problems for $(b-a)|\lg\varepsilon| = 1$

Class	Stiff ratio
Stiff at limit	$S = \mathcal{O}(10)$
Midle stiffly	$S = \mathcal{O}(10^2)$
Strong stiffly	$\mathcal{O}(10^3) \leq S < \mathcal{O}(10^5)$
Extreme stiffly	$\mathcal{O}(10^6) \leq S < \mathcal{O}(10^8)$
Pathological stiffly	$\mathcal{O}(10^9) \leq S$

In physics and chemestry, any real system which is modelated by an ordinary differential equation system with components supposing some evolutions in different dials of time, also referred to as *time constants*, is treated as a stiff system. A large time constant determines the stiff character. The components of practical interest are those with a slow variation. Integrating the problem with an explicit method, the stepsize must be of small constant order.

Example 6. A nonlinear system is stiff if at least for one point $t_s \in [a,b]$ the associated linear system $y'(t) = (\partial f/\partial y)(t_s, y(t_s))y(t),\ t \geq t_s$ is stiff.

The meaning of the sign $\gg$ in the above mentionated inequalities is relative to the integration interval length and the error tolerance. An initial value problem can not be classified as stiff one, also when the eigenvalues are of very different values, if the integration interval can be compared with the transitory stage, or if the error tolerance ε is not very small.

Definition 1.3. The *stiff ratio* of the system (1.13) is

$$S(A,a,b,\varepsilon) = \max_{i=1,\ldots,N} |\lambda_i|(b-a)\lg\frac{1}{\varepsilon}, \tag{1.14}$$

where a, b are the extreme values of the integration interval and ε is the error tolerance.

Depending on the stiff ratio the problems can be classified like in Table 1.1.

In chemestry dynamics we can often meet a stiff ratio of order $\mathcal{O}(10^6)$.

Definition 1.4. An ordinary differential equation system $y' = f(t,y)$, numerical integrated on the interval $[a,b]$, starting from the initial value $y(a) = y_a$, is *stiff* if the function f has a LIPSCHITZ's constant very large relative to the interval length, and the error tolerance.

Remark. For a system which satisfies only the condition of dissipativity with the constant μ, the stiff or nonstiff character is decided comparing the value μ, the interval length, and the error tolerance. Note that the dissipativity constant satisfies the inequality

$$\mu \geq \sup_{y\in R^N} \mu[(\partial f/\partial t)(t,y)] \tag{1.15}$$

where $\mu[A] = \sup_{x\in\mathbf{R}^N} < x, Ax > / < x, x >$ is the logarithmic norm of the matrix A.

The principal characteristics of the stiff problems are (AITKEN[1]):

- the exact solutions are stable in the sense that small perturbations in the initial values are followed only by small perturbations in the exact solutions;
- trying to solve the problem with the standard methods we get some strict restrictions on the stepsize from the stability conditions.

A problem necessarly must have these two characteristics to be declared a stiff one.

Chapter 2

LINEAR MULTISTEP FORMULAE

The linear multistep formulae are the first methods which were proposed for solving the stiff problems (Curtis, 1952). Due to the stiff character, this class of formula suffers a variety of restrictions. This fact explains the development of some new class of methods, like the nonlinear multistep methods, which are related to the multistep linear formulae.

The fundamentals of the modern theory of stability have been establish by Dahlquist (1956). From 1971, when Gear has published his test results on some special linear multistep methods, these are often used in the stiff computation.

2.1 Formula

Let $(t_0, t_1, \ldots, t_M)$ a division of the integration interval with constant stepsize $h = t_{i+1} - t_i$, $i = 0, \ldots, M-1$. A linear multistep formula is represented by an iterative process of the followings form:

$$\boxed{\sum_{i=0}^{k} \alpha_{k-i} y_{n-i} - h \sum_{i=0}^{k} \beta_{k-i} f_{n-i} = 0,} \tag{2.1}$$

where $f_{n-i} = f(t_{n-i}, y_{n-i})$ and $M \geq n \geq k$. If $(\alpha_0^2 + \beta_0^2)(\alpha_k^2 + \beta_k^2) \neq 0$, then k is *the formula stepnumber.*

The formula is an *explicit* one if $\beta_k = 0$, else it is an *implicit* one. If $\alpha_k = 0$, $\beta_k \neq 0$, then the formula is an *extended* one.

At the step n, knowing the $(k-1)^{\text{th}}$ previous approximate values, we get the k^{th} value by writing the formula in the form

$$\alpha_k y_n - h\beta_k f_n = \text{constant},$$

In the explicit case, when $\beta_k = 0$, solving this equation means only a simple division of a constant expression with α_k. In the implicit case and a N-dimensional system

function f we must, generally, solve a nonlinear system of N equations, so that it is necessary to use an iterative procedure for this special problem.

The classical procedure for solving the implicit equation or the system of implicit equations is the *predictor-corrector scheme*. The predictor linear multistep formula, usually an explicit one, give a starting value for the iterative process for solving the implicit corrector formula. The approximation delivered by the predictor formula is improved applying the corrector formula. The standard iterative process for solving the implicit corrector equation is the method of simple iterations, in the nonstiff case, and the NEWTON's method, in the stiff case.

The formula (2.1) can be applied only if we know the values $y_0, \ldots, y_{k-1}$, referred to as the *starting values*. These are delivered by a procedure which is independent from the basic formula, a *starting procedure*. Naturally, the stepnumber of such a procedure is smallest that of the multistep formula. In almost all present numerical codes, the starting procedure includes some onestep methods.

The starting procedure must satisfies two conditions: the *boundless* and the *compatibility*. The boundless refers to the existence of a supperior limit for the approximation values, the procedure outputs, an uniformely bound relative to a small variation interval for the stepsize. The starting procedure is compatible if the outputs values converge to a fixed value, when $h \to 0$. These two condition ensure the existence and the unicity of the numerical solution. The linear multistep formula has an *unique solution* for a specific variation interval for the stepsize and for any bounded and compatible starting procedure.

2.2 Consistency and order

The consistency of a linear multistep formula makes sure that the numerical solution (the theoretical one) converges to the exact solution when $h \to 0$. In this condition, we can find a stepsize h for which the approximation error is under some tolerance level.

2.2.1 Definitions

Let the linear operator associated with a linear multistep formula [86]

$$L := \sum_{j=0}^{k} \alpha_j H^j - h \sum_{j=0}^{k} \beta_j H^j \frac{d}{dt}, \tag{2.2}$$

where H is the displacement operator defined as $Hy(t) = y(t+h)$.

Definition 2.1. L has *the precision degree* p, if $L \equiv 0$ for any t^j, $j = 0, \ldots, p$, and p is the maximum integer that safisfies this property.

The local error of a linear multistep formula is the difference

$$y(t_n) - \overline{y}_n,$$

where $y(t_n)$ is the exact value of the solution at t_n, and $\overline{y}_n$ is the approximate value of the solution at t_n, produced applying the formula with $y_{n-i} = y(t_{n-i}),\ i = 1, \ldots, k$. If the system function f is differentiable continous, then the local error can be approximated by (MIRANKER[85])

$$y(t_n) - \overline{y}_n \approx \left(\alpha_k I - h\beta_k \frac{\partial f}{\partial y}(t_n, \mu) \right)^{-1} L(y(t_n - hk)),$$

where μ is a value between $y(t_n)$ and y_n. Thus, we can justify the following definition.

Definition 2.2. A given linear multistep formula has p *order* if the associated linear operator has a precision degree p. The linear multistep formula is *consistent* if its order satisfies $p \geq 1$.

2.2.2 Algebraic conditions

Let the expansion in TAYLOR's series of the linear multistep operator

$$L(y(t)) = \sum_{i=0}^{\infty} c_i h^i y^{(i)}(t).$$

A precision degree p involves the relations $c_i = 0,\ i = 0, \ldots, p$. The values c_i can be expressed in terms of α_i and β_i.

The algebraic system, which is equivalent with the condition of p-order, is the following:

$$\sum_{i=0}^{k} \alpha_i = 0, \qquad \sum_{i=0}^{k} \frac{i^j}{j!} \alpha_{k-i} - \sum_{i=0}^{k} \frac{i^{j-1}}{(j-1)!} \beta_{k-i} = 0, \quad j = 1, \ldots, p \tag{2.3}$$

Moreover,

$$c_{p+1} = \frac{1}{(p+1)!} \left[\sum_{i=0}^{k} \alpha_{k-i} i^{p+1} - (p+1) \sum_{i=0}^{k} \beta_{k-i} i^p \right] \neq 0.$$

For a linear multistep with k steps, the inequation

$$\boxed{p \leq 2k.}$$

is referred to as **the order barrier**. Consequently, there is an only one implicit formula with k and of maximum order $2k$ and an only one explicit formula with k steps and maximum order $2k - 1$.

Remark. A p order linear multistep formula exactly solves an equation with a polynomial solution of lowest degree than p.

The methods with k steps are generated by polynomials pairs (ρ, σ) with real coefficients, where

$$\rho(\xi) := \sum_{j=0}^{k} \alpha_j \xi^j, \quad \sigma(\xi) := \sum_{j=0}^{k} \beta_j \xi^j. \tag{2.4}$$

Suppose that $(\rho, \sigma) = 1$.

If E is the translation operator, specifical for the numerical process, $Ey_n = y_{n+1}$, then the linear formula with k steps can be rewritten in the form

$$\rho(E)y_{n-k} = h\sigma(E)f_{n-k}.$$

The consistency condition can be algebraic expressed in the relationships

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1) \neq 0. \tag{2.5}$$

The root $\xi = 1$ of the polynomial ρ is referred to as the *principal root*, and the others, as the *external roots*.

The p order condition can be also set in the terms of the pair (ρ, σ):

$$\rho(e^h) - h\sigma(e^h) = \mathcal{O}(h^{p+1}), \qquad h \to 0.$$

Using the transformation $e^h \mapsto x$ we get an equivalent equality

$$\frac{\rho(x)}{\ln x} - \sigma(x) = \mathcal{O}((x-1)^p), \qquad x \to 1.$$

Note

$$r(z) := \left(\frac{z-1}{2}\right)^k \rho\left(\frac{z+1}{z-1}\right) = \sum_{j=0}^{k} a_j z^j,$$

$$s(z) := \left(\frac{z-1}{2}\right)^k \sigma\left(\frac{z+1}{z-1}\right) = \sum_{j=0}^{k} b_j z^j. \tag{2.6}$$

The p order conditions can be expressed as the WIEDLUNG's relationships:

$$a_k = 0, \ \frac{a_{k-1}}{2} = b_k, \ \frac{a_{k-2}}{2} = b_{k-1}, \ a_{k-3} = b_{k-2} + \frac{b_k}{3}, \ \frac{a_{k-4}}{2} = b_{k-3} + \frac{b_{k-1}}{3},$$

$$\frac{a_{k-j}}{2} = b_{k-j-1} + \frac{1}{3}b_{k-j-3} + \frac{1}{5}b_{k-j-5} + \cdots + \begin{cases} \frac{1}{j}b_k, & j \text{ odd} \\ \frac{1}{j-1}b_{k-1}, & j \text{ even} \end{cases}, j \leq \min(p,k) \tag{2.7}$$

Suppose $a_i = b_i = 0, i < 0, i > k$. Therefore,

$$a_i = 2\sum_{j \geq 0} \frac{b_{2j+1+i}}{2j+1}, \qquad i = k, k-1, \ldots, k-p.$$

The consistency condition can be rewritten:

$$\rho(1) = a_k = 0, \qquad \rho'(1) = \sigma(1) = b_k = a_{k-1}/2 \neq 0 \tag{2.8}$$

The difference between two methods with the same order can be overtake based on the error constant. The local error of a p order formula can be represented in the form

$$y(t_n) - y_n = c^* h^{p+1} y_n^{(p+1)} + \mathcal{O}(h^{p+2}). \tag{2.9}$$

Definition 2.3. The value $c^* = -c_{p+1}/\sigma(1)$ is referred to as the *error constant* of the linear multistep formula (ρ, σ) of $p > 1$ order.

The error constant can be easily computed (MIRANKER[86]):

$$c^* = \lim_{x \to 1} \frac{\ln x - \rho(x)/\sigma(x)}{(x-1)^{p+1}}.$$

The p order *accuracy* supposes a p order of consistency and a small error constant (relative to the stepsize).

2.3 Stability

There are many examples of consistent numerical formulae which not converge to the exact solution when $h \to 0$. The reason of this phenomenon is the accumulation of the calculus errors. The most refined is the integration division, the computational effort increases and it is possible that, at $h \to 0$, the accumalated errors introduce some perturbations which are bigest than the exact solution values.

Let consider an initial value problem and the system with some perturbated initial condition. A good numerical method for solving the initial value problem do not increase without limits the ratio between the actual difference of the two numerical solution and the initial difference (the perturbation of the initial condition), in any condition for the stepsize. This property is referred to as *stability* of the numerical method.

2.3.1 Zero-stability

The stability study of the linear multistep methods is based on the concept of *root condition.*

Definition 2.4. A polynomial satisfies the *root condition* if all its roots are inside the complex closed unitary disk and those from the boundary are not roots of the derivative polynomial.

Theorem 2.1. *[85] If a linear multistep method (ρ, σ) is stable, then $\rho(\xi)$ satisfies the root condition.*

The polynomial ρ is referred to as the *stability polynomial.* If ρ satisfies the root condition, we say that the formula is *zero-stable.*

The reverse implication is true when the linear multistep formula is also consistent.

Counterexample.[52] Let the consistent formula

$$y_n + 4y_{n-1} - 5y_{n-2} = h(4f_n + 2f_{n-1}).$$

The stability polynomial has the root -5. If we apply the formula to the differential equation $y' = -y$, with $y(0) = 1$, we get the equation

$$y_n + 4(1+h)y_{n-1} - (5-2h)y_n = 0.$$

The solution of the this equation has the form

$$y_n = ax_1^n(h) + bx_2^n(h),$$

where $x_i(h),\ i = 1, 2$ are the roots of the equations

$$x^2 + 4(1+h)x - (5-2h) = 0$$

i.e. $x_1(h) = 1 - h + \mathcal{O}(h^2),\ x_2(h) = -5 - \mathcal{O}(h)$. Note that $|y_n| \to \infty,\ n \to \infty$.□

The stability concept is extremely important for the convergence of the iterative process. *The approximations produced by a consistent linear multistep formula converge to the exact solution when* $h \to 0$ *iff the formula is zero-stable,* i.e.

$$\text{Stability} \longleftarrow \begin{cases} \text{Consistency} \\ \\ \text{Root condition} \end{cases} \begin{matrix} \nwarrow \\ \longrightarrow \\ \swarrow \end{matrix} \text{Convergence}$$

In the construction of efficient numerical codes, the DAHLQUIST *'s barriers* are very important. These are the theorems which impose some restrictions on the stable formula class. The *first* DAHLQUIST *'s barrier* is the following theorem.

Theorem 2.2. *[53] The order of a consistent and zero-stable linear multistep formula can not exceed the following values*

$$p \leq \begin{cases} k+1, & \text{if } k \text{ is odd,} \\ k+2, & \text{if } k \text{ is even.} \end{cases}$$

Remarks:

(1) The stability barrier is more restrictive than the order barrier. This fact is valid only for the linear multistep formula, not also for the onestep methods like the RUNGE-KUTTA's process.

(2) Moreover, HAIRER and WANNER [53] have proved that, $p \leq k$, if $\beta_k/\alpha_k \leq 0$ (particulary also for the explicit formulae).

If all the secondary roots of ρ are inside of the complex unitary circle, the formula is *strongly stable.* A formula is strongly stable iff $r(z)$ is an HURWITZ polynomial, i.e. it has all the roots inside the complex half-plane $\mathrm{Re}\, z > 0$ (that implies $a_j \geq 0,\ j = 0, \ldots, k$).

2.3.2 Absolute stability. Stability domain. A-stability

The absolute stable methods are particular cases of zero-stable methods. The idea is to request the decrease of the secondary numerical solution with the increase of the stepnumber. The study is made on the scalar test equation $y' = \lambda y,\ \mathrm{Re}\,\lambda < 0$. The absolute stability is the minimum property that must be imposed on any numerical integration method. *The behaviour of the method on the test equation is a prevision model of the behaviour in the case of some nonlinear systems.*

Let the equation produced applying the method with constant stepsize h to the test equation:

$$\sum_{j=0}^{k}(\alpha_j - q\beta_j)y_{n-k+j} = 0, \quad q = \lambda h \tag{2.10}$$

The *characteristic equation* can be get from this formula replacing the values y_{n-k+j} with ξ^j:

$$\sum_{j=0}^{k}\alpha_j\xi^j - q\sum_{j=0}^{k}\beta_j\xi^j = 0, \quad q = \lambda h. \tag{2.11}$$

The polynomial $\rho - q\sigma$ is referred to as the *characteristic polynomial* of the linear multistep formula.

Definition 2.5. A method is *absolute stable* for a value $q = h\lambda$ if, for the value q, the characteristic polynomial satisfies the root condition.

Theorem 2.3. *[45] A consistent linear multistep formula is absolute stable at $q = h\lambda \in C$ iff all the solutions $\{y_n\}$ of the equation (2.11), produced applying the formula to the test equation $y' = \lambda y$ with constant stepsize h, are bounded when $n \to \infty$.*

The proof of the direct implication is similar to the one of the implication stability $\rightarrow$ zero-stability. The reverse implication makes use of the consistency hypothesis.

Counterexample. Let the same formula like in the last section and the equation $y' = \lambda y$, with $y(0) = 1$, $\mathrm{Re}\,\lambda < 0$. The solution of the difference equation has the form $y_n = a[1 - h\lambda + \mathcal{O}(h^2\lambda^2)]^n + b[-5 - \mathcal{O}(h\lambda)]^n$ where $|h\lambda| < 1$. Thus $|y_n| \to \infty$ since the second values converge to infinity. On an other hand, also $|1 - h\lambda| > 1$, $\forall h > 0$.

We can associate to any linear multistep formula a stability domain.

Definition 2.6. The *absolute stability domain* of a linear multistep formula is the set

$$A = \{h\lambda | \text{ applying the formula to the problem } y' = \lambda y, \lambda \in \mathbf{C},\ y(t_0) = y_0,$$
$$\text{with the constant stepsize } h > 0, \text{ we get some approximate values}$$
$$y_n \text{ which } y_n \to 0, \text{ when } n \to \infty\}.$$

The stability domain is the set of the complex points for which the linear multistep formula is absolute stable.

The boundary of the stability domain lies in some parts of the curve[53]

$$q(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}, \qquad \theta \in [0, 2\pi)\ . \tag{2.12}$$

The stability domains of the classical methods, like the ADAMS-BASHFORTH, ADAMS-MOULTON, NYSTRÖM or MILNE-SIMPSON's formulae, are almost all bounded. The stability requirement leads to the condition that the values $|h\lambda_i|$, where λ_i are the eigenvalues of the JACOBIAN's matrix at the moment t_n, must be inside the stability domain. A large value $|\lambda_i|$ implies a small value h. If the stability domain is unbounded, for example includes the complex left half-plane, then the condition $h\lambda_i \in A$ does not restricted the stepsize.

Definition 2.7. A formula is *A-stable* if the absolute stability domain includes the complex left half-plane, i.e. those values z for which $\mathrm{Re}\,(z) < 0$. A *method is A-stable* if it is based on an A-stable formula and, moreover, the implicit equation produced applying the formula at the test equation is exactly solved.

Remark. The method notion is more general than that of a formula, scheme or process. Especially in the stiff case, a method must includes also a procedure for solving the implicit equations. We can also include in a method the step variation techniques and the order variation tehniques, the estimating error algorithms or the algorithm for stiff detection. In the following sections, we refer to a method only as a formula, scheme or process in an implementation with a constant stepsize.

Let

$$\mathbf{C}_- = \{z \in \mathbf{C} \,|\, \mathrm{Re}\,(z) < 0\}. \tag{2.13}$$

Thus, the condition of A-stability can be expressed by the inclusion

$$\boxed{\mathbf{C}_- \subset A,}$$

and that of zero-stability by

$$\boxed{0 \in A.}$$

For an onestep method, the A-stability condition is derived from a property of the *stability function* $R(q) = y_{n+1}/y_n$. An onestep method is A-stable if the stability function satisfies $|R(z)| \leq 1,\ \forall \mathrm{Re}\,(z) < 0$..

Also, if the linear multistep formula has an unbounded stability domain, it is possible to have an undesirable property:

$$|y_{n+1}/y_n| \to 1, \qquad h\lambda \to -\infty.$$

For example, the trapezoidal rule, which is A-stable, has this property. Thus, the fastly decreasing components of the exact solution are represented in the numerical solution by some slowly decreasing components, even by some oscilatory ones, if $y_{n+1}/y_n < 0,\ \forall n > 0$.

Definition 2.8.

(i) A linear multistep formula is *stable at infinity* if $\lim_{h\lambda\to-\infty} |y_{n+1}/y_n| = 0$, where y_n is the numerical solution of the scalar test equation.

(ii) The formula is *L-stable* or *stiffly A-stable* if it is A-stable and it is stable at infinity.

(iii) A linear multistep formula is *strongly A-stable* if it is A-stable and there is a value $w > 0$ such that

$$\sup_{h\lambda<-w} |R(h\lambda)| < 1$$

2.3.3 Sufficient conditions for A-stability

A practical method to decide that a formula is A-stable consists in applying the following lemma proposed by MIRANKER [85].

Lemma 2.1. *If $\xi_i, i = 1, \ldots, k$ are the roots of the characteristic equation, then the following statements are equivalent:*

(i) the method is A-stable;

(ii) if $\operatorname{Re} q < 0$, *then* $|\xi_i| \leq 1, \quad i = 1, \ldots, k$;

(iii) $\forall \xi : |\xi| > 1$, $\operatorname{Re} q(\xi) = \operatorname{Re}[\rho(\xi)/\sigma(\xi)] \geq 0$ *holds for* $|\xi| > 1$.

Note W the exterior of the closed unitary complex disk. Using the lemma 2.1 we can prove the following proposition[86].

Proposition 2.1.

(i) The linear multistep method is A-stable only if the roots σ_i of $\sigma(\xi)$ safisfy $|\sigma_i| \leq 1, \ i = 1, \ldots, k$;

(ii) The linear multistep formula is A-stable iff $\operatorname{Re}[\rho(\xi)/\sigma(\xi)] \geq 0, \forall \xi \in \overline{W}$.

(iii) If a root of $\sigma(\xi)$ lies on the unitary circle and is not a simple root, then the linear multistep formula is not A-stable.

Applications:

(1) The trapezoidal rule is A-stable since $\rho(\xi) = \xi - 1$, $\sigma(\xi) = \frac{1}{2}(\xi + 1)$, $\operatorname{Re} q(\xi) = (|\xi|^2 - 1)/|\xi + 1|^2$and thus $\operatorname{Re} q(\xi) > 0$, for all $|\xi| > 1$, and the root of σ from the unitary circle is a singular one.

(2) The EULER's implicit rule is A-stable since $\rho(\xi) = \xi - 1$, $\sigma(\xi) = \xi$, $\operatorname{Re} q(\xi) = (|\xi|^2 - \operatorname{Re}\xi)/|\xi + 1|^2 > 0, \ \forall |\xi| > 1$;

(3) An example of A-stable and consistent method for arbitrary stepnumber $k \geq 1$ is the following:

$$y_n - y_{n-k} = \frac{kh}{2}(f_n + f_{n-k}).$$

Since $\rho(\xi) = \xi^k - 1$, $\sigma(\xi) = \frac{k}{2}(\xi^k + 1)$, the roots of $\sigma(\xi)$ are separated ones and lie on the unitary circle, and $\operatorname{Re} q(\xi) = \frac{k}{2}(|\xi|^{2k} - 1)\big/|\xi^k + 1|^2 > 0, \ \forall |\xi| > 1$.

Proposition 2.2. *[86] If*

(i) the roots σ_i of the σ polynomial satisfy $|\sigma_i| < 1, \ i = 1, \ldots, k$, and

(ii) $u(\xi) = \operatorname{Re} q(\xi) \geq 0$ for ξ on the unitary circle,

then the linear multistep formula is A-stable.

Applications:

(1) Let the class of onestep methods which depend on the parameter a (referred to as a-rules)

$$y_{n+1} - y_n = h[(1-a)f_{n+1} + af_n].$$

We apply the last proposition. The unique root of the σ polynomial is $\sigma_1 = -a/(1-a)$ with $|\sigma_1| < 1$ iff $a < 1/2$. An other hand, $u(e^{i\theta}) = |\sigma(e^{i\theta})|^{-2}P(e^{i\theta})$, where $P(e^{i\theta}) = (1-2a)(1-\cos\theta)$. Then $P(e^{i\theta}) \geq 0$ iff $a \leq 1/2$. Consequently, the method is A-stable if $a < 1/2$. Note that we have only an implication. A counterexample is that of the trapezoidal rule which is an A-stable formula with $a = 1/2$.

(2) The two step formulae of order two can be represented in the form

$$(1-c)y_n + 2cy_{n-1} + (-1-c)y_{n-2} = h[bf_n + (2-a-b)f_{n-1} + af_{n-2}],$$

where $c = a - b$. From Proposition 2.2, the inequalities

$$b - a > 0, \ -1 + a + b > 0,$$

are sufficient conditions for A-stability.

Corollary 1. *If*

(i) $\mathrm{Re}(s_i) < 0, i = 1, \ldots, k$, *where* s_i *are the roots of the polynomial* $s(z)$,

(ii) $\mathrm{Re}(r(z)/s(z)) \geq 0, \forall z : \mathrm{Re}(z) = 0$,

then the linear multistep formula is A-stable.

The request condition (i) can be treated with the ROUTH-HURWITZ's theory. More details can be found in [52], [45] and [96].

Note

$$\sum_{i=0}^{k} e_i y^{2i} = \mathrm{Re}\,[r(iy)s(-iy)].$$

A sufficient condition for (ii) is $e_i \geq 0$. The coefficients e_i can be computed from the following relationships built up in the conditions $a_i = b_i = 0, \ i < 0, \ i > k$:

$$e_i = \sum_{j \geq 0} (-1)^{i+j} a_j b_{2i-j}, \qquad i = 0, \ldots, k$$

2.3.4 Necessary conditions for A-stability

The class of A-stable linear multistep formulae is not very large, since such a formula must be implicit and with an order of accuracy lowest than three.

Theorem 2.4. *(ENRIGHT[38]) A linear multistep formula, with a stability domain which includes the negative real half-axis, must be an implicit one.*

Corollary 1. *An explicit linear multistep formula cannot be A-stable.*

This result shows that it is necessary to solve at each step an implicit equation or a system of implicit equations (a great computational overhead), which can be also a source for suplimentary errors.

The following theorem, due to DAHLQUIST, gives a full characterization of the restrictions impose by the A-stability condition. It is referred to as the **second DAHLQUIST's barrier**.

Theorem 2.5. *(DAHLQUIST[34]) The order of an A-stable linear multistep formula can not exceed two. The trapezoidal rule is the two order A-stable formula with the lowest error constant* $c^* = 1/12$.

Corollary 1. *For a linear multistep formula*

$$\boxed{\mathbf{C}_- \subset A \Rightarrow \ p \leq 2.}$$

The are many *strategies to overtake* the second DAHLQUIST's barrier. First of all, we can apply some RUNGE-KUTTA's process. These must also be implicit when we impose the condition of A-stability, and at each step we must solve a system of $q \times N$ implicit equations, where q is the number of intermediary stages (compared with only N equations in the case of a linear multistep formula). The advantage of using such process is that of the identity between the barrier order and the barrier imposed by the A-stability condition on the accuracy order. The onestep method are not the subject of this study. More details can be found in [53], [16] or [96].

The principal ways to break the DAHLQUIST's second barrier are:

1. weaking the stability condition: stiff stability, A(α) stability and so on;
2. using the implicit RUNGE-KUTTA's process;
3. using the high derivatives of the solution:
   - *a*) multiderivative multistep formulae;
   - *b*) multiderivative RUNGE-KUTTA's process;
4. using some nonlinear interpolators:
   - *a*) nonlinear multistep methods;
   - *b*) rational RUNGE-KUTTA's process;
5. adding some new stages, extradivision points or future points:
   - *a*) cyclic linear multistep methods and composed methods;
   - *b*) hybrid methods (extradivision points) and multistep RUNGE-KUTTA methods;
   - *c*) extended methods (future points);
6. reducing the approximation error:
   - *a*) exponential fitting;
   - *b*) extrapolation methods;
7. exploiting the special structure of the system: stiff separability.

(2.14)

These ways for breaking the barrier are presented in the following sections.

2.3.5 A(α)-stability

Small variations of the system JACOBIAN's matrix produce small variations of its eigenvalues λ_i. By such a variation, the values $h\lambda_i(t)$ cover only a small part of the complex left half-plane. The covered region can be delimited by two lines with the slopes less than one. For the problems for which the JACOBIAN's matrix eigenvalues are not included in the imaginary axis, it is justified the request of absolute stability on a smallest region than the entire left complex half-plane.

Definition 2.9. A linear multistep formula is *A(α)-stable*, with $0 < \alpha < \pi/2$, if all the solutions of the difference equation, produced applying the formula to the scalar test equation $y' = \lambda y$, converge to zero when $n \to \infty$, for any fixed stepsize $h > 0$, and for any $\lambda \neq 0$ so that

$$h\lambda \in \mathbf{C}_\alpha = \{q \,|\, |\arg(-q)| < \alpha, q \neq 0\}$$

holds.

The angle α is referred to as the *stability angle of the method.*
Formally, A(α)-stability means

$$\boxed{\mathbf{C}_\alpha \subset A.}$$

The algebraic conditions can be deduced similarly to the case of the A-stability property:

- a linear multistep formula is A(α)-stable if, for any $q \in \mathbf{C}_\alpha$, all the roots of the characteristic equation satisfy $|\xi_i| < 1,\ i = 1, \ldots, k$.
- a linear multistep formula is A(α)-stable if all the roots s_i of the polynomial $s(z)$ satisfy $\mathrm{Re}\, s_i \leq 0, i = 1, \ldots, k$;
- a linear multistep formula is A(α)-stable iff $r(z)/s(z)$ lies in the exterior of the set $\mathbf{C}_\alpha,\ \forall z :\ \mathrm{Re}\, z > 0$;

The most important *properties* of the above notion are the followings:

1. A(α)-stability $\Rightarrow$ A(β)-stability for $0 < \beta < \alpha$;
2. A($\pi/2$)-stability $\Leftrightarrow$ A-stability;
3. an explicit formula can not be A(α)-stable, since $\mathbf{C}_\alpha$ includes the negative real half-axis;
4. **it is possible to overtake the second DAHLQUIST's barrier**, since are some A(α)-stable formula for $0 \leq \alpha < \pi/2$ and $k = p = 3$ or $k = p = 4$.

Theorem 2.6. *(GRIGORIEFF, SCHROLL[51]) Let $\alpha < \pi/2$. Then for any k a positive integer number there is a A(α)-stable linear multistep formula with k steps and of order $p = k$.*

The limit case is $\alpha = 0$.
Remarks:

(a) An explicit formula can not be A(0)-stable, since the stability domain of a such formula must includes the negative real half-axis;

(b) The trapezoidal rule is the only one A(0)-stable formula with $p \geq k+1$.

For a linear multistep A(0)-stable formula $\alpha_j \geq 0$ (or $\alpha_j \leq 0$) and $\beta_j \geq 0$ (or $\beta_j \leq 0$) hold for all $j = 0, \ldots, k$ (LINIGER[81]).

2.3.6 A^0-, A_0-, $A_\infty-$stability

Building up these notions, the idea is to weak the condition of A-stability.

Definition 2.10. Let a linear formula with k steps.

(i) The formula is A^0-*stable* if, applying it to the test equation with λ a negative real number, we get some approximations y_n with the property $y_n \to 0$ when $n \to \infty$, for any constant stepsize h (NEVANLINNA[90]).

(ii) The formula is A_0-*stable* if the characteristic polynomial satisfies the root condition for any negative real value q (CRYER[32]).

(iii) The formula is A_∞-*stable* if $s(z)$ is an HURWITZ polynomial (LINIGER[81]).

The most important properties are the followings:

(1) A^0-stability $\Leftrightarrow$ $(-\infty, 0] \subset A$.

(2) A(0)-stability $\Rightarrow$ A_0-stability $\Rightarrow$ A^0-stability.

(3) The formula is A_0-stable iff all the roots of the polynomial $r(z) - qs(z)$ are inside the set $\mathbf{C}_-$ for all $q \in (-\infty, 0)$ (CRYER[33]).

(4) If the formula is A_0-stable, then $a_j b_j \geq 0,\ j = 0, \ldots, k$.

(5) The formula is A_0-stable iff $\forall z \in \mathbf{C}_+, r(z)/s(z)$ is regular and positive, and for a z value with $z = \operatorname{Im} z$, $r(z)\overline{s(z)} > 0$.

(6) If the formula is A_∞-stable, then $b_j > 0,\ j = 1, \ldots, k$ and the order $p \leq k$ (LINIGER[81]).

(7) The formula is A_0-stable if it is A_∞-stable, $a_0 \geq 0$, and the polynomial

$$-\frac{1}{y}\operatorname{Im}\left[r(iy)s(-iy)\right] = \sum_{j=0}^{k-1} f_j \eta^j, \quad y \in \mathbf{R}, \quad \eta = y^2 \geq 0,$$

has only real positive roots. The last condition holds if $f_j > 0,\ j = 0, \ldots, k-1$.

(8) The formula is A(0)-stable iff (JELTSCH[66]):

(i) it is A_0-stable;

(ii) all the roots ξ_i of the polynomial $\sigma(\xi)$ with $|\xi_i| = 1$ satisfy $\sigma'(\xi_i) \neq 0$ and $\mathrm{Re}\,(\rho(\xi_i)/(\xi_i\sigma'(\xi_i)) > 0$;

(iii) all the roots ξ_j of the polynomial $\rho(\xi)$ with $|\xi_j| = 1$ satisfy $\mathrm{Re}\,(\sigma(\xi_j)/(\xi_j\rho'(\xi_j)) > 0$.

Application. The methods with k steps, the order k, and the polynomial $s(x) = (x+d)^k$ (the CRYER's formulae), are A_0-stable iff $d \geq 2^{k+1}$.

Counterexample. The formula

$$y_n - y_{n-1} = \frac{h}{4}(f_n + 2f_{n-1} + f_{n-2})$$

is A_0-stable, but not A(0)-stable.

2.3.7 Stiff stability

Let a linear multistep formula and ξ_i, the roots of the characteristic polynomial. Then

$$S = \{\lambda \in \mathbf{C} | |\xi_i(\lambda)| < |\xi_1(\lambda)|, i = 2, \ldots, k\} \tag{2.15}$$

is referred to as the *relative stability domain*, and also as the *star region*.

Definition 2.11. (GEAR[45]) A linear multistep formula is *stiffly-stable* with respect to the D, a, θ parameters, if

$$R_1 \cup R_2 \subset A, \qquad R_3 \subset S \tag{2.16}$$

where

$$R_1 = \{\lambda \in \mathbf{C} | \,\mathrm{Re}\,\lambda < -D\},\ R_2 = \{\lambda \in \mathbf{C} |\ -D \leq \mathrm{Re}\,\lambda \leq -a, |\mathrm{Im}\,\lambda| < \theta\},$$

$$R_3 = \{\lambda \in \mathbf{C} | \,|\mathrm{Re}\,\lambda| < a, |\mathrm{Im}\,\lambda| < \theta\}.$$

The reason of this notion is the following. For the test equation the exact solution increases at one step with $e^{h\lambda}$. If $h\lambda = u + iv$, the amplification term is e^u. If $u < -D < 0$, the solution component decreases very fast. Since the solution is very small, we are not interesed in accuracy of that, only in its stability. In the neighbourhood of the origin, the accuracy becames important. The relative stability is a necessary condition for accuracy. If $|v| > \theta$, at least $\theta/(2\pi)$ complete cycles are performed at each step. These variation is very hard to be followed if θ is large, and we can use a reasonable stepsize (relative to the integration interval length).

The classical *examples* of stiffly stable formulae are the backward difference formulae of maximum order (GEAR's formulae).

The stiff-stability notion is very important in breaking the second DAHLQUIST's barrier. For any arbitrary stepnumber k we can build a stiffly stable linear formula with k steps.

Theorem 2.7. *(CRYER[32]) For any value $D > 0$ and any positive integer k there is a linear formula with k steps, stiffly stable, and of k order, with*

$$R_1 \subset A,$$

where A is the stability domain of the formula.

The proof is based on the fact that for any D we can get a value d such that the relationship from the theorem holds for the CRYER's method generated by $\sigma(\xi) = (\xi + d)^k$ and the condition of maximum order.

The relationships between the stiff-stability notion and the above mentionated notion can be expressed in the following diagram:

$$\text{A-stability} \Rightarrow \text{stiff stability} \Rightarrow \text{A}(\text{arctg}(\theta/D))\text{-stability} \Rightarrow$$
$$\Rightarrow \text{A}(0)\text{-stability} \Rightarrow A_0\text{-stability}$$

Couterexample. The method generated by the pair of polynomials

$$\rho(\xi) = \xi^3 - \xi^2/2 - 1/2, \quad \sigma(\xi) = 3\xi^3/2 - \xi^2 + 3\xi/2$$

is A(0)-stable but not stiffly stable.

A reverse implication supposes some supplimentary conditions:

Proposition 2.3. *(JELTSCH[66]) A linear multistep formula is stiffly stable iff*

(i) it is A_0-stable;

(ii) the roots of the polynomial $\rho(\xi)/(\xi - 1)$ are inside the unitary circle;

(iii) all roots ξ_i of the polynomial $\sigma(\xi)$ for which $|\xi_i| = 1$ satisfy $\sigma'(\xi_i) \neq 0$ and $\rho(\xi_i)/(\xi_i\sigma'(\xi_i)) \in \mathbf{R}_+$.

Note that COOKE[30] has established some sufficient conditions for which a consistent linear multistep methods is stiffly stable.

2.4 Convergence

The consistency condition supposes that the numerical process exactly models the solution of the initial value problem when the stepsize $h \to 0$. One principal supposition consists in the fact that the calculus is carried without any perturbations. The more general notion of convergence makes reference to the same phenomenon, but allows the intervention of some small perturbations.

Any numerical method must satisfy the condition of convergence to demonstrate its efficiency in the integration of an initial value problem, since the convergence supposes a control of the error produced by the numerical process.

The global error (accumulated error) of a numerical method is

$$e_n = y(t_n) - v_n, \tag{2.17}$$

where $y(t_n)$ is the exact solution, and v_n is the approximate solution at t_n, produced by the numerical code. The boundary of this error leads us to the notion of convergence.

Definition 2.12. A method is *convergent* if

$$\lim_{h\to 0} \max_n \|e_n\| = 0 \tag{2.18}$$

holds for all the initial value problems with the standard conditions for the solution unicity.

2.4.1 Global error of an onestep method

Let the onestep method

$$y_{n+1} = y_n + h\Phi(h, y_n, y_{n+1}).$$

Note the *global discretization error*

$$\overline{e}_{n+1} = y(t_{n+1}) - y_{n+1}, \tag{2.19}$$

produced by the accumulation of the *local discretization errors,*

$$\overline{l}_{n+1} = y(t_{n+1}) - \overline{y}_{n+1},$$

where

$$\overline{y}_{n+1} = y(t_n) + h\Phi(h, y(t_n), \overline{y}_{n+1}).$$

The onestep method is of p *order of consistency*, if p is the greatest integer which satisfies

$$\overline{l}_{n+1} = \mathcal{O}(h^{p+1}), \qquad h \to 0.$$

If the method is stable, then there is a constant value K for which

$$\|\overline{y}_{n+1} - y_{n+1}\| \le K\|\overline{y}_n - y_n\|, \quad \overline{y}_n = y(t_n)$$

holds, and

$$\|\overline{e}_{n+1}\| \le K\|\overline{e}_n\| + \|\overline{l}_{n+1}\| \le K\|\overline{e}_n\| + \overline{\rho},$$

where $\overline{\rho}$ is the superior bound of the local discretization error. Thus, the global discretization error $\overline{e}_{n+1}$ satisfies the inequality

$$\|\overline{e}_{n+1}\| \le K^{n+1}\|\overline{e}_0\| + \overline{\rho}\sum_{i=0}^{n} K^i. \tag{2.20}$$

If $K \le 1$, i.e. the method is *contractive*, there is not a significant propagation of the global error, only an accumulation of the local error, and, if $\overline{e}_0 = 0$, then $\|\overline{e}_{n+1}\| \le \overline{\rho}/(1-K)$.

If $K = 1 + C_0 h$, $h \in (0, h_0]$, where h_0, C_0 are independent of the initial value problem, then

$$\|\overline{e}_{n+1}\| \le \overline{\rho}\sum_{i=0}^{n}(1+C_0h)^i = \overline{\rho}\frac{(1+C_0h)^{n+1}-1}{C_0h} \le$$

$$\leq \begin{cases} \overline{\rho} h^{-1} t_{n+1}, & C_0 \leq 0 \\ \overline{\rho} h^{-1} C_0^{-1} (e^{C_0 t_{n+1}} - 1), & C_0 > 0 \end{cases}, \quad h \in (0, h_0]$$

when $\|\overline{e}_0\| = 0$.

If there is not a constant value C_0 so that $K = 1 + C_0 h$, then the method is not stable.

If we approximate the exact solution with the perturbated numerical values v_n, the propagation of these perturbations can be bounded in the same manner like in the propagation of the global discretization error.

Note the *global numerical error*

$$\tilde{e}_{n+1} = y_{n+1} - v_{n+1}, \tag{2.21}$$

produced by the accumulation of the *local numerical errors*

$$\tilde{l}_{n+1} = v_{n+1} - \tilde{y}_{n+1},$$

where

$$\tilde{y}_{n+1} = v_n + h\Phi(h, v_n, \tilde{y}_{n+1}).$$

We get

$$\|\tilde{e}_{n+1}\| \leq K\|\tilde{e}_n\| + \|\tilde{l}_{n+1}\|, \quad \tilde{\rho} \geq \|\tilde{l}_{n+1}\|, \quad \|\tilde{e}_{n+1}\| \leq K^{n+1}\|\tilde{e}_0\| + \tilde{\rho}\sum_{i=0}^{n} K^i. \tag{2.22}$$

Hence, the *global error*

$$e_{n+1} = y(t_{n+1}) - v_{n+1} = \overline{e}_{n+1} + \tilde{e}_{n+1}, \tag{2.23}$$

satisfies

$$\|e_{n+1}\| \leq K^{n+1}(\|\overline{e}_o\| + \|\tilde{e}_0\|) + (\overline{\rho} + \tilde{\rho})\sum_{i=0}^{n} K^i. \tag{2.24}$$

In practice, we suppose that $\tilde{\rho} \ll \overline{\rho}$.

2.4.2 Convergence of the linear multistep formula

Note $y_h(t)$ the interpolation function of the approximate values, produced applying a linear multistep formula at an integration interval division with constant stepsize h.

Definition 2.13. A linear multistep formula is *convergent* if

$$\|y(t) - y_h(t)\| \to 0, \qquad h \to 0,\ t \in [t_0, t_0 + T],$$

holds for any initial value problem with the standard conditions for the unicity of the exact solution, independent from the starting values, which only satisfy

$$\|y(t_0 + ih) - y_h(t_0 + ih)\| \to 0, \qquad h \to 0,\ i = 0, \ldots, k-1.$$

The algebraic conditions for convergence are the zero-stability and the consistency.

Theorem 2.8. *[52] If a linear multistep formula is convergent, then*

(i) is zero-stable;

(ii) is consistent.

The reverse implication is based on the transformation of a linear multistep formula into an onestep method. Let $Y_i = (y_{i-1}, y_{i-2}, \ldots, y_{i-k})^T,\ i \geq 0$,

$$A = \begin{pmatrix} -\alpha_{k-1}/\alpha_k & -\alpha_{k-2}/\alpha_k & \ldots & -\alpha_0/\alpha_k \\ 1 & 0 & \ldots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{pmatrix}$$

and the function Ψ, uniquely defined by the relationship,

$$\Psi = \frac{1}{\alpha_k} \sum_{j=0}^{k-1} \beta_j f(t_{i-j}, y_{i-j}) + \frac{\beta_k}{\alpha_k} f\left(t_i, h\Psi - \sum_{j=0}^{k-1} \frac{\alpha_{k-j}}{\alpha_k} y_{i-j}\right).$$

Then the linear multistep formula can be rewritten in the form of an onestep formula [53]:

$$Y_{i+1} = (A \odot I) Y_i + h\Phi(t_i, Y_i, h),$$

where $A \odot I$ is the KRONECKER's tensorial product, and

$$\Phi(t_i, Y_i, h) = (e_1 \odot I)\Psi(t_i, Y_i, h), \qquad e_1 = (1, 0, \ldots, 0)^T.$$

Theorem 2.9. *[53] The zero-stability and the consistency imply the convergence of the linear multistep formula.*

Moreover, the global error can be estimated when the formula is A-stable or only A(α)-stable. For a linear system

$$y' = Ay + g(t).$$

the following result holds.

Proposition 2.4. *[53] If the linear multistep formula has p order of accuracy, is also A(α)-stable, stable at infinity, and the system matrix A has the property that there is a matrix T so that $T^{-1}AT = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_N)$, where $|\arg(-\lambda_i)| \leq \alpha,\ i = 1, \ldots, N$, then there is a constant value M, which depends only on the formula, so that for any stepsize $h > 0$, the global error satisfies*

$$\|y(t_n) - y_n)\| \leq M\|T\|\|T^{-1}\| \left(\max_{0 \leq j \leq k} \|y(t_j) - y_j\| + h^p \int_{t_0}^{t_n} \|y^{(p+1)}(\zeta)\| d\zeta \right) \qquad (2.25)$$

More generally is the following result concerning the nonlinear systems.

Proposition 2.5. *[53] If the method is strongly A-stable, of p order of accuracy and the initial value problem satisfies the disipativity condition*

$$Re < f(t,u) - f(t,v), u - v > \leq \mu \|u - v\|^2,$$

then there is a constant value $C_0 > 0$ so that

$$\|y_n - y(t_n)\| \leq C(\max_{0\leq j\leq k} \|y_j - y(t_j)\| + h \max_{0\leq j\leq k} \|f(t_n, y_j) - y'(t_j)\| + Mh^p, \qquad (2.26)$$

holds if $h\mu \leq C_0$, where C and M depend on the method, and, for $\mu > 0$, also on $t_n - t_0$. Moreover, M depends on the exact solution derivatives of a given order.

2.4.3 Convergence of the discretization methods

CHARTRES and STEPLEMAN[27] present a theory of convergence, consistency and stability in which the definitions are more general than the above mentionated ones.

Definition 2.14. *A discretization method* is a set $(F, \{F_h\}, \{E_h\}, H, X, Y)$ where $F : X \to Y$, $F_h : X \times E_h \to Y$ for each $h \in H$, $0 \in E_h$, in H we have defined the convergence $h \to 0$, but $0 \notin H$, and Y is a linear normed space.

In this definition we have note:

- X: the date space, the set of pairs (y_0, f);
- Y: the function space which includes the exact solution and the interpolation function of the dicrete values;
- $F : X \to Y$ maps each pairs $(y_0, f) \in X$ into the exact solution $y \in Y$;
- E_h: the set of the N_h components of the perturbations produced at each step $i = 0, \ldots, N_h$ of stepsize h (practical, is a finite dimensional space);
- $F_h : X \times E_h \to Y$ maps each pairs of an element from X and a specific perturbation into the interpolation function of the numerical discrete values;
- H: the set of the admissible stepsizes.

Suppose, for each method and each $h \in H$, the existence of a function $\phi_h : E_h \to R$, with $\phi_h(0) = 0$ for any $h \in H$. Then by $e_h \to 0$, we mean $\phi_h(e_h) \to 0$, and by $(h, e_h) \to 0$, $h \to 0$ and $\phi_h(e_h) \to 0$.

Definition 2.15.

(i) A discretization method is *convergent* at x if $\lim_{(h,e_h)\to 0} F_h(x, e_h) = F(x)$. The order of convergence is p if $\|F_h(x, e_h) - F(x)\| = \mathcal{O}(h^p)$, for $\phi_h(e_h) = \mathcal{O}(h^p)$.

(ii) A discretization method is *consistent* at x if there is a specifical perturbation e_h^c for which $h \in H$ such that $\lim_{h\to 0} \phi_h(e_h^c) = 0$, and $\lim_{h\to 0} F_h(x, e_h^c) = F(x)$. The order of consistency is p if $\|F_h(x, e_h^c) - F(x)\| = \mathcal{O}(h^p)$ for $\phi_h(e_h^c) = \mathcal{O}(h^p)$.

(iii) A discretization method is *stable* at x if $\lim_{(h,e_h)\to 0} \|F_h(x,e_h) - F_h(x,0)\| = 0$.

Theorem 2.10. *[27] A discretization method is convergent iff it is consistent and stable.*

Application. The EULER's explicit rule applied to an initial value problem on the integration interval [0,1] is a discretization method $(F, \{F_h\}, \{E_h\}, H, X, Y)$ with

$$X = \{(y_0, f) | f : [0,1] \times R \to R, \text{ for which } y' = f(t,y),\ y(0) = y_0,$$

$$\text{has an unique solution } y\},$$

$$Y = C^1[0,1], \quad F(f, y_0) = y, \quad E_h = R^{N_h+1},$$

$$N_h = \frac{1}{h}, \quad H = \{1, 1/2, 1/3, \ldots\}, \quad F_h((f,y_0),e) = u$$

where

$$u = I_h(u_0, u_1, \ldots, u_{N_h})$$

$$u_0 = y_0 + e_0, \quad u_n = u_{n-1} + hf(t_{n-1}, u_{n-1}) + e_n,\ n = 1, 2, \ldots, N_h \tag{2.27}$$

$$e = (e_0, e_1, \ldots, e_{N_h}) \in E_h$$

If $u^* = I_h(u_0^*, \ldots, u_{N_h}^*)$ is the numerical solution associated to $e_n = 0, n = 0, \ldots, N_h$ and f satisfies the LIPSCHITZ's condition in the dependent variable, then

$$\|u_0 - u_0^*\| \le e_0, \quad \|u_n - u_n^*\| \le (1 + hL)\|u_{n-1} - u_{n-1}^*\| + \|e_n\|,$$

where L is the LIPSCHITZ's constant, and, therefore,

$$\|u_n - u_n^*\| \le e^L \sum_{r=0}^{n} \|e_r\|, \quad n = 0, \ldots, N_h.$$

Hence, the most weakest topology in E_h, for which the method is stable, is based on $\phi_h(e) = \sum_{r=0}^{N_h} |e_r|$. Note TE_n the local discretization error in t_n. Then

$$y(t_n) = y(t_{n-1}) + hf(t_{n-1}, y(t_{n-1})) + TE_n$$

holds and the characteristic perturbation e_h^c is the vector $(0, TE_1, \ldots, TE_{N_h})$. In the proposed topology $\phi_h(e_h^c) = \sum_{n=1}^{N_h} TE_n$. The method is consistent if $\phi_h(e_h^c) \to 0$, when $h \to 0$. Since $TE_n = \mathcal{O}(h^2)$, if f is differential continous and has a bounded derivative, the method is convergent.

2.5 Linear multistep formulae for stiff problems

There are three ways to build-up a linear multistep formula:

- by numerical integration; for example, we can mention the ADAMS-BASHFORTH, ADAMS-MOULTON, NYSTRÖM, MILNE-SIMPSON or NEWTON-CÔTES formulae;

- by numerical diferentiation; for example, we can mention the backward differentiation formulae;
- by combination of the coefficients of some known formulae.

A bounded stability domain is a characteristical property for the methods of the above mentionated first way. Consequently, these formulae are not suitable for the integration of stiff problems. Most adequate are the following methods:

- *the backward differentiation formulae* (short noted BDF). For $k \leq 6$, the methods with k steps and maximum order k, known as GEAR's formulae, are stiffly stable;
- *the methods of* ADAMS *'s type.* Note C_q the ADAMS-MOULTON's method of $q+1$ order with q steps, and the generating polynomials

$$\rho_q(\varepsilon) = \varepsilon^q - \varepsilon^{q-1}, \quad \sigma_q(\varepsilon) = \sum_{j=0}^{q} \gamma_j \varepsilon^{q-j}(\varepsilon - 1)^j, \quad \gamma_j = (-1)^j \int_0^1 C_u^j du,$$

  Let $b_1, \ldots, b_k$ some real values. We can associated to those values a method of $q+1$ order with $q+k$ steps (RODABAUGH, THOMPSON[118])

$$C(b_1, \ldots, b_k) = \sum_{m=1}^{k} C_{q+m} + \left(1 - \sum_{m=1}^{k} b_m\right) C_q; \tag{2.28}$$

- *the α-methods* with the general formula

$$y_n - \alpha y_{n-1} + (\alpha - 1) y_{n-2} = h \sum_{i=0}^{k} \beta_i f_{n+i-k}, \quad 0 < \alpha < 2. \tag{2.29}$$

  For $\alpha = 1$ we get the ADAMS's methods. ROCKSWOLD[117] proves that there are such methods which are A_0-stable, for the orders $p = 1, \ldots, 4$;
- *the* CRYER *'s methods* are defined by the polynomial $\sigma(\varepsilon) = (\varepsilon + d)^k$, $d \in (-1, 1)$ and the condition of maximum order of accuracy (CRYER[32]). JELTSCH[67] proves that the CRYER's method with k steps is $A(0)$-stable and stiffly stable if $-1 < d \leq -1 + 2/(1 + 2^{k+1})$, where $k = 1, \ldots, 7$;
- *the* GRIGORIEFF-SCHROLL *'s methods* [51] are computed using the polynomial $\sigma(\varepsilon) = (\varepsilon + d)^{k-1}(\varepsilon + c)$ and the condition of maximum order. The numerical tests have proved that, for some values d and c, the methods are stiffly stable for $k \leq 12$ steps.

2.6 Backward differentiation formulae (BDF)

The BDF are the first formulae used in the integration of the stiff problems.

2.6.1 Formula

A backward differentiation formula with k steps, BDF_k, can be represented by the iterative process

$$\boxed{\sum_{j=0}^{k} \alpha_j y_{n+j-k} = h\beta_k f_n,} \tag{2.30}$$

where $(\alpha_k^2 + \beta_k^2)\alpha_0 \neq 0$.

The formula of maximum order is the result of a numerical differentiation procedure. The basic idea is to determine an interpolation polynomial for which

$$P(t_{n+i-k}) = y_{n+i-k}, \quad i = 0, \ldots, k, \qquad P'(t_n) = f(t_n, y_n)$$

holds. The polynomial is built-up using the backward differences. Note $\bigtriangledown$ the backward differentation operator. Thus, the GEAR's formula with k steps is the following process

$$\sum_{j=1}^{k} \frac{1}{j} \bigtriangledown^j y_n = h f_n. \tag{2.31}$$

2.6.2 Particular cases

- The GEAR's formulae [45] or BDF with maximum order (equal to the stepnumber), are stiffly stable for $k \leq 6$ steps.
- The VARAH's formulae [120] reduce with an unity the method order, in the favour of a greatest stability angle than that of GEAR's formulae.
- The CASH's extended BDF [22] suppose an implementation in a predictor-corrector scheme: the predictor formula is one of the GEAR's methods, and the corrector formula is an extended one:

$$(P)\ \sum_{j=0}^{k} \overline{\alpha}_j y_{n+j-k} = h\overline{\beta}_k f_n,\ (C)\ \sum_{j=0}^{k} \alpha_j y_{n+j-k} = h\beta_k f_n + h\beta_{k+1} f_{n+1}. \tag{2.32}$$

  The scheme has $k+1$ order of accuracy. The purpose of using this scheme is to improve the maximum order of the stiffly stable formulae, and to maximize the stability angle. One step consists in the following stages:

  1. compute $\overline{y}_n$ by (P);
  2. compute $\overline{y}_{n+1}$ by (P);
  3. evaluate $\overline{f}_{n+1}$;
  4. compute y_n by (C).

2.6.3 Similar methods

- *The split linear multistep methods* have been proposed by CASH [24]. A split formula with k steps has the form

$$\sum_{j=0}^{k} \alpha_j y_{n+j-k} - h \sum_{j=0}^{k-1} \beta_j f_{n+j-k} = h\theta f(t, \overline{y}_n) + h(\beta_k - \theta) f(t_n, y_n) \qquad (2.33)$$

  where $\overline{y}_n$ is a value generated by a predictor formula of the form

$$(P) \qquad \sum_{j=0}^{k} \overline{\alpha}_j y_{n+j-k} = h \sum_{j=0}^{k} \overline{\beta}_j f_{n+j-k}$$

  CASH has studied the effect on the stability of splitting the BDFs.

- *The* DILL*'s formulae* [36] lie in the iterative process

$$\sum_{j=0}^{k} \alpha_j y_{n+j-k} = h(\beta_k f_n + \beta_{n-u} f_{n-u}), \quad \beta_k = 1, \quad u \in \{1, \ldots, k\} \qquad (2.34)$$

  The purpose of building these formula is to increase the order of stiffly stable formulae.

2.6.4 Stability

A suggestive proof for the stiff-stability of GEAR's formulae is the study of the stability domain plots. The boundary of the stability domain for the k-step formula lies on the curve

$$\theta \in [0, 2\pi) \longmapsto z = \sum_{j=1}^{k} \frac{1}{j} (1 - e^{-i\theta})^j.$$

Thus, for $k = 1$,

$$A = \{z \,|\, |z - 1| \geq 1\},$$

and for $k = 2$,

$$\partial A = \{z \,|\, \mathrm{Re}\, z = 3/2 - 2\cos\theta + 1/2 \cos 2\theta,\ \theta \in [0, 2\pi)\}.$$

hold. The stability domains for $k < 7$ are plotted in Figure 2.1.

The coefficients and the parameters of stiff-stability are given in Table 2.1.

Note that the first GEAR's formula is the implicit EULER's rule.

HAIRER and WANNER [55] have proved that the GEAR's formulae are unstable for $k \geq 7$.

The establishment of the stepsize variation interval is a very important problem. The stiff stability property can be lose applying an unadequate stepsize variation technique (GEAR et al.[46] [47]).

The VARAH's formulae [120] increase the order of the stiffly stable formulae with two units. The formulae are $A(\alpha)$-stable for $k \leq 12$ steps and $p \leq 8$ order. The stability angles are mentionated in Table 2.2.

The CASH's extended schemes have two qualities:

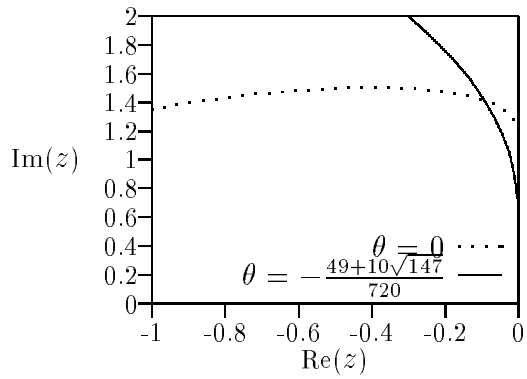

Figure 2.1: Stability domains of the GEAR's formulae

Table 2.1: The GEAR's formula coefficients,the stability angle and the stiff parameter D

$k=p$	α_6	α_5	α_4	α_3	α_2	α_1	α_0	β_k	α	D
1						1	-1	1	90	0
2					1	$-\frac{4}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	90	0
3				1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$	$\frac{6}{11}$	86.03	0.083
4			1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$	$\frac{12}{25}$	73.35	0.667
5		1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$\frac{12}{137}$	$\frac{60}{137}$	51.84	2.327
6	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$	$\frac{60}{147}$	17.84	6.05

1. the maximum order of a stiffly stable formula is 9, greatest than 6, the maximum order for GEAR's formulae (see Table 2.3);

2. it proves the possibility to break the DAHLQUIST's second barrier, since the scheme of $p=3$ and $p=4$ order are A-stable.

An other example of breaking the second barrier is the following CASH's split backward differentiation formula:

$$(P)\quad \overline{y}_n - \left(\frac{18}{11}+\frac{5}{2}\theta\right) y_{n-1} + \left(\frac{9}{11}+4\theta\right) y_{n-2} - \left(\frac{2}{11}+\frac{3}{2}\theta\right) y_{n-3} = \left(\frac{6}{11}-\theta\right) h\overline{f}_n,$$

$$(C)\ y_n - \frac{18}{11} y_{n-1} + \frac{9}{11} y_{n-2} - \frac{2}{11} y_{n-3} = \left(\frac{6}{11}-\theta\right) h f_n + \theta h f(t_n, \overline{y}_n). \qquad (2.35)$$

The scheme has $p=3$ order and it is L-stable for $\theta \in (-1.1, -0.29)$. In the same manner, we can built-up some split backward differentiation formulae with $k \geq 3$ steps, in the idea to improve the stability angle of the GEAR's formulae. CASH proves (by plotting the stability domains boudaries) that:

Table 2.2: The stability angles for the VARAH's formulae

k	7	8	9	10	11	12
p	5	6	7	7	8	8
α	72	55	29	39	2	30

Table 2.3: The stability angle for the CASH's extended BDF schemes

k	1	2	3	4	5	6	7	8
p	2	3	4	5	6	7	8	9
α	90	90	90	87.61	80.21	67.73	48.82	19.98

- the 4-step formula of 4^{th} order is L(α)-stable with $\alpha > 89.999°$ for a fixed θ value,
- the 5-step formula of 5^{th} order is L(α)-stable with $\alpha > 85°$ for a fixed θ value,
- the 6-step formula of 6^{th} order is L(α)-stable with $\alpha > 71°$ for a fixed θ value.

An other example is the split scheme based on the trapezoidal rule:

$$(P) \qquad \overline{y}_n - y_{n-1} = \theta h f(t_n, \overline{y}_n) + (1-\theta) h f(t_{n-1}, y_{n-1})$$

$$(C) \qquad y_n - y_{n-1} = \theta h f(t_n, y_n) + \frac{1}{2} h f(t_{n-1}, y_{n-1}) + h\left(\frac{1}{2} - \theta\right) f(t_n, \overline{y}_n)$$

The scheme is stable at infinite if $\theta = 1 \pm 1/\sqrt{2}$, A-stable if $\theta \geq 1/4$, and we get the minimum error when $\theta = 1 - 1/\sqrt{2}$.

The stiff stability of the DILL's formulae have been also proved in [36] plotting the stability domains.

The stability results concerning the above mentionated methods are concentrated in the following lemma.

Lemma 2.2. *Let k the stepnumber of a backward differentiation formula.*

(i) The GEAR*'s formulae are A-stable for $k \leq 2$ and stiffly stable for $k \leq 6$. The maximum order for a stiffly stable formula is 6 (*GEAR*[45]).*

(ii) The VARAH*'s formulae are stiffly stable for $k \leq 12$. The maximum order for a stiffly stable formula is 8 (*VARAH*[120]).*

(iii) The CASH*'s extended BDF schemes are A-stable for $k = 1, 2$ and stiffly stable for $k \leq 8$. The maximum order for a stiffly stable formula is 9 (*CASH*[22]).*

(iv) The CASH*'s split BDF schemes are A-stable for $k = 3$ and some θ values, and stiffly stable for $k \leq 6$ and some θ values. The maximum order for a stiffly stable formula is 6 (*CASH*[24]).*

(v) The DILL *'s class of formulae of $p = k - 1$ order includes some methods of 7 and 8 order which are stiffly stable. The maximum order for a stiffly stable formula is 8 (*DILL*[36]).*

Chapter 3

GENERALIZED LINEAR MULTISTEP FORMULAE

Building up some linear multistep formula of high order for the stiff problem integration we can follow two directions:

1. using the high derivatives of the exact solution;
2. adding some supplimentary stages, extradivision points or future points, steping in a large field of new methods like:
   (a) block methods;
   (b) hybrid methods;
   (c) extended methods.

In the last section we have mentioned an example of an extended method.

The low order linear multistep formula proposed in the last chapter can be improved, from the point of view of the error, applying one of the following methods:

- exponential fitting;
- extrapolation.

In the following sections we discuss these ways.

3.1 Second derivative linear multistep formulae

The linear multistep formula uses only the derivative of the exact solution $y'(t) = f(t, y(t))$. A second derivative multistep formula uses also the second derivative of the exact solution $y"(t) = d(f(t, y(t))/dt$.

The motivation for using such a formula is due to the fact that a linear multistep method for integrating the stiff problems must include a scheme for solving the implicit equation systems. The usual predictor-corector scheme can not be applied since the algebraic condition for convergence ($\|h(\partial f/\partial y)\|$ to be a small value) is contradictory

with the stiff character of the problem. The iteration schemes for solving the implicit equations are based on some modifications of the standard NEWTON's iteratives, which are exactly for linear systems. These methods avoid the difficulty of convergence, but, unfortunately, they ask the evaluation of the JACOBIAN's matrix. Therefore, we can use this matrix explicitly in the linear formula.

3.1.1 Formula

Note $g(t) := y"(t) = [(\partial f/\partial t) + (\partial f/\partial y)f](t, y(t))$. The iterative process, which represents the formula, is

$$\boxed{\sum_{i=0}^{k} \alpha_{k-i} y_{n-i} = h \sum_{i=0}^{k} \beta_{k-i} f_{n-i} + h^2 \sum_{i=0}^{k} \gamma_{k-i} g_{n-i}.} \tag{3.1}$$

3.1.2 Particular cases

- The ENRIGHT's formulae are selected based on three requests:

  1. stability at infinity: a single nonzero coefficient γ_i, respectivelly γ_k;
  2. some reasonable stability properties in a neighbourhood of the origin of the complex plane: the polynomial ρ is the same like that of the ADAMS's formulae. The iterative process is

$$y_n = y_{n-1} + h \sum_{i=0}^{k} \beta_i f_{n+j-k} + h^2 \gamma_k g_n. \tag{3.2}$$

  3. maximum order: the coefficients are unique defined (ENRIGHT [38]),

$$y_n = y_{n-1} + h f_n + h \sum_{j=1}^{k} \frac{\nabla^j f_n}{j} \left(\sum_{i=j}^{k} c_i \right) + h^2 g_n \left(\sum_{i=0}^{k} c_i \right),$$

  where

$$c_i = \int_0^1 \frac{(s-1)^2 s(s+1)(s+2)\cdots(s+i-2)}{i!} ds.$$

- The second derivative backward differentiation formulae (SDBDF) [53] have the coefficients derived from the condition of maximum order:

$$\sum_{j=1}^{k} \frac{1}{j} \left(\sum_{i=1}^{k} \frac{1}{i} \right) \nabla^j y_n = \left(\sum_{i=1}^{k} \frac{1}{i} \right) h f_n - \frac{h^2}{2} g_n. \tag{3.3}$$

3.1.3 Similar methods

- The CASH's second derivative extended schemes [21] consists in ENRIGHT's method as the predictor formula and an extended method as the corrector formula:

$$(P) \quad \sum_{j=0}^{k} \overline{\alpha}_j y_{n+j-k} = h \sum_{j=0}^{k} \overline{\beta}_j f_{n+j-k} + h^2 \sum_{j=0}^{k} \gamma_j g_{n+j-k},$$

$$(C) \quad \sum_{j=0}^{k} \alpha_j y_{n+j-k} = h \sum_{j=0}^{k+1} \beta_j f_{n+j-k} + h^2 \sum_{j=0}^{k} \gamma_j g_{n+j-k}. \tag{3.4}$$

- A modification of the above mentionated extended formulae is also presented by CASH [18] as a composed second derivative multistep scheme, for autonomous initial value problems with $f' = Jf$, where $J = (\partial f / \partial y)$ is the JACOBIAN's matrix:

$$(P) \quad y_n - y_{n-1} = h \sum_{j=0}^{k} \overline{\alpha}_j f_{n+j-k} + hJ(\sum_{j=0}^{k} \overline{\beta}_j y_{n+j-k} + h \sum_{j=0}^{k} \overline{\gamma}_j f_{n+j-k}),$$

$$(C) \quad y_n - y_{n-1} = h \sum_{j=0}^{k+1} \alpha_j f_{n+j-k} + hJ(\sum_{j=0}^{k} \beta_j y_{n+j-k} + h \sum_{j=0}^{k} \gamma_j f_{n+j-k}). \tag{3.5}$$

  The idea of this modification consists in the fact that the linear system matrix, which results applying the NEWTON's method, is more simple than the corresponding one resulting from (3.4).

- The CASH's extended second derivative extended backward differentiation schemes [23] are particular cases of the following predictor-corrector schemes:

$$(P) \quad \sum_{j=0}^{k} \overline{\alpha}_j y_{n+j-k} = h \sum_{j=0}^{k} \overline{\beta}_j f_{n+j-k} + h^2 \sum_{j=0}^{k} \overline{\gamma}_j g_{n+j-k},$$

$$(C) \quad \sum_{j=0}^{k} \alpha_j y_{n+j-k} = h \sum_{j=0}^{k+1} \beta_j f_{n+j-k} + h^2 \sum_{j=0}^{k+1} \gamma_j g_{n+j-k} \tag{3.6}$$

  One scheme step consists in the following stages:

  1. compute $\overline{y}_n$ by (P);
  2. compute $\overline{y}_{n+1}$ by (P);
  3. evaluate $f(\overline{y}_{n+1})$ and $g(\overline{y}_{n+1})$;
  4. compute y_n by (C);
  5. estimate error using the difference $y_n - \overline{y}_n$.

Moreover, the methods are chosen so that they be zero-stable and stable at infinity. The two classes studied by CASH are:

$$\text{Class 1} \quad \begin{aligned} &(P) && y_n - y_{n-1} = h\sum_{i=0}^{k} \overline{\beta}_i f_{n+i-k} + h^2 \overline{\gamma}_k g_n, \\ &(C) && y_n - y_{n-1} = h\sum_{i=0}^{k+1} \beta_i f_{n+i-k} + h^2(\gamma_k g_n + \gamma_{k+1} g_{n+1}). \end{aligned}$$

$$\text{Class 2} \quad \begin{aligned} &(P) && y_n - y_{n-1} = h\sum_{i=0}^{k} \overline{\beta}_i f_{n+i-k} + h^2 \overline{\gamma}_k g_n, \\ &(C) && y_n - y_{n-1} = h\sum_{i=0}^{k+1} \beta_i f_{n+i-k} + h^2 \gamma_k g_n. \end{aligned}$$

- The SKELL-KONG's formulae [53] consists in some combinations of the ADAMS's type methods

$$(AMF^{(k+1)}) \qquad -y_n + y_{n-1} + h\sum_{i=1}^{k} \beta_i f_{n+i-k} = 0,$$

used in the integration of nonstiff problems, with the BDF, used in the integration of stiff problems,

$$(BDF^{(k)}) \qquad -\sum_{i=0}^{k} \alpha_i y_{n+i-k} + h f_n = 0,$$

in the form

$$(AMF^{(k+1)}) - \gamma^{(k)} h J (BDF^{(k)}) = 0.$$

3.1.4 Order

The algebraic conditions of p order are the followings:

$$\sum_{j=0}^{k} \alpha_j j^q = q\sum_{j=0}^{k} \beta_j j^{q-1} + q(q-1)\sum_{j=0}^{k} \gamma_i i^{q-2}, \qquad 0 \le q \le p.$$

More simplified,

$$\begin{gathered} \rho_1(1) = 0, \qquad \rho_2(1) - \sigma_1(1) = 0, \\ \rho_{j+1}(1) - j\sigma_j(1) - j(j-1)\delta_{j-1}(1) = 0, \qquad j = 1, \ldots, p, \\ c_{p+1} = \rho_{p+2}(1) - (p+1)\sigma_p(1) - p(p+1)\delta_p(1) \neq 0, \end{gathered}$$

where

$$\rho_1(\xi) = \sum_{i=0}^{k} \alpha_i \xi^i, \quad \sigma_1(\xi) = \sum_{i=0}^{k} \beta_i \xi^i, \quad \delta_1(\xi) = \sum_{i=0}^{k} \gamma_i \xi^i,$$

$$\rho_{j+1}(\xi) = \xi\rho_j'(\xi), \quad \sigma_{j+1}(\xi) = \xi\sigma_j'(\xi), \quad \delta_{j+1}(\xi) = \xi\delta_j'(\xi).$$

The error constant is

$$c^* = \frac{c_{p+1}}{\sigma(1)} =$$

$$= \frac{1}{\sigma(1)(p+1)!}\left[\sum_{j=0}^{k}\alpha_j j^{p+1} - (p+1)\sum_{j=0}^{k}\beta_j j^p - (p+1)p\sum_{j=0}^{k}\gamma_j j^{p-1}\right]$$

Lemma 3.1.

(i) The ENRIGHT*'s formula with k steps has the order $p = k+2$.*

(ii) The SDBDF with k steps has the order $p = k+1$.

(iii) The SKELL-KONG*'s formula with k steps has the order $p = k+1$ for any $\gamma^{(k)}$.*

(iv) If the predictor formula has the order $p = k+2$ and the corrector formula, the order $p = k+3$, then the order of the CASH*'s extended second derivative extended backward differentiation scheme is $p = k+3$.*

3.1.5 Stability

The stability domain of a second derivative linear multistep formula is the set

$$A = \{z \in \mathbf{C} \mid \text{ the polynomial } \rho - z\sigma - z^2\delta \text{ satisfies the root condition}\},$$

where $\delta(\xi) = \sum_{i=0}^{k}\gamma_i\xi^i$.

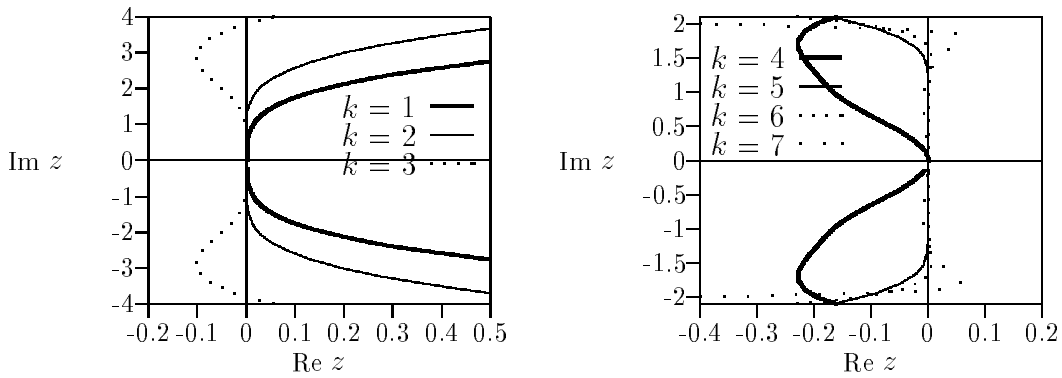

Figure 3.1: Stability domains for the ENRIGHT's formulae

The characteristic equation is

$$\sum_{i=0}^{k}(\alpha_i - z\beta_i - z^2\gamma_i)\xi^i = 0, \quad z = h\lambda.$$

For $\xi = e^{i\theta},\ \theta \in [0, 2\pi)$ we get two root curves which describe the boundary of the stability domain.

The DAHLQUIST's second barrier is moved by two units. The greatest order of an A-stable second derivative linear multistep formula is four. This fact is a corollary of a more general theorem, which show us that the maximum order of an A-stable linear multistep formula is twofold than the maximum degree of the solution derivatives used in the formula (the DANIEL-MOORE's conjucture [53]).

The ENRIGHT's formulae break the DAHLQUIST's barrier since the first two are A-stable with $p > 2$. The coefficients are given in Table 3.1, the stability angle in Table 3.2, and the stability domains in Figure 3.1.

Table 3.1: ENRIGHT's formula coefficients and the order of accuracy

k	β_{k-6}	β_{k-5}	β_{k-4}	β_{k-3}	β_{k-2}	β_{k-1}	β_k	γ_k
1						$\frac{1}{3}$	$\frac{2}{3}$	$-\frac{1}{6}$
2					$-\frac{1}{48}$	$\frac{5}{12}$	$\frac{29}{48}$	$-\frac{1}{8}$
3				$\frac{7}{1080}$	$-\frac{1}{20}$	$\frac{19}{40}$	$\frac{307}{540}$	$-\frac{19}{180}$
4			$-\frac{17}{5760}$	$\frac{1}{45}$	$-\frac{41}{480}$	$\frac{47}{90}$	$\frac{3133}{5760}$	$-\frac{13}{32}$
5		$\frac{41}{25200}$	$-\frac{529}{40320}$	$\frac{373}{7560}$	$-\frac{1271}{10080}$	$\frac{2837}{5040}$	$\frac{317731}{604800}$	$-\frac{863}{10080}$
6	$-\frac{731}{725760}$	$\frac{179}{20160}$	$-\frac{5771}{161280}$	$\frac{8131}{90720}$	$-\frac{13823}{80640}$	$\frac{12079}{20160}$	$\frac{247021}{483840}$	$-\frac{275}{3456}$

Table 3.2: Stability angles of the ENRIGHT's formulae

k	1	2	3	4	5	6	7
α	90	90	87.88	82.03	73.10	51.95	37.61
p	3	4	5	6	7	8	9

For solving the implicit equations we need some starting values. If we use an explicit predictor formula, the scheme is equivalent to an explicit method (LAMBERT[74]). CASH[25] has proved that the ENRIGHT's onestep method can be applied in the following scheme:

(P) compute $y_{n-1}^{(0)}, y_n^{(0)}$ with the trapezoidal rule, starting with a value produced by the EULER's explicit rule,

(C) compute the approximate value of the solution at t_n with the modified ENRIGHT's formula

$$y_n - y_{n-1} = \frac{h}{3}(2f_n + f_{n-1}) - \frac{\delta^2 y_n^{(0)}}{6}$$

where $\delta y_n = y_{n+1} - y_n$, and the starting value $y_n^{(0)}$,

This scheme is A-stable and of third order (an example for breaking the DAHLQUIST's barrier).

There are many improvements proposed for the ENRIGHT's formulae:

- ENRIGHT [39] give up to an order unit in favour to solve more easy the linear system produced applying the simplified NEWTON's iterations. The system matrix has the form

$$(I - h\beta_k J - h^2\gamma_k J^2) = (I - h\beta_k J)^2 \quad \Rightarrow \quad \gamma_k = -(\beta_k/2)^2$$

  The other coefficients are determinated from the maximum order request. For $k \leq 8$ the condition system has two distinct solution, depending on β_k. The choise of one of these solutions is due to the numerical tests proposed by ENRIGHT.

- CHAKRAVARTI and KANEL [26] propose a modification in the idea to improve the order of the stiffly stable formulae. For stability at infinity, the ENRIGHT's condition that $\sigma(x) = \gamma_k x^k$ is successively replaced by the conditions

$$\begin{array}{ll} (a) & \sigma(x) = \gamma_k x^{k-1}(x+a), \\ (b) & \sigma(x) = \gamma_k x^{k-2}(x+a)(x+b), \\ (c) & \sigma(x) = \gamma_k x^{k-3}(x+a)(x+b)(x+c). \end{array}$$

  The numerical tests on the influence of the parameters a, b, c prove the existence of some stiffly stable methods for

$$(a) \quad 3 \leq k \leq 8, \qquad (b) \quad 3 \leq k \leq 10, \qquad (c) \quad 3 \leq k \leq 12$$

The two classes of CASH's schemes [23] include some stiffly stable methods for $k \leq 6$. The maximum order for such a stiffly stable scheme is 9. The ENRIGHT's methods are used only as predictor formulae. In Table 3.3 we have mention the stability angles.

Table 3.3: Stability angles for the CASH's scheme from the second class

k	1	2	3	4	5	6
p	4	5	6	7	8	9
α	90	90	90	89	87	83

The coefficients of the first four SDBDF are mentionated in Table 3.4. The stability regions are plotted in Figure **??**. In Table 3.5 we mention the stability angles.

The SKELL-KONG's formulae have the same stability properties as the ENRIGHT's formulae, but they are more easy to be implemented. The parameter $\gamma^{(k)}$ is choosed to maximize the stability angle (see Table 3.6).

The conclusions about the stability of the proposed particular cases of second derivative formulae are concentrated in the following lemma.

Lemma 3.2.

(i) The Enright*'s formulae are A-stable for $k \leq 2$ and stiffly stable for $k \leq 7$. The maximum order of a stiffly stable formula is $p = 8$ (*Enright *[38]).*

(ii) The Cash*'s schemes from the two class of extended second derivative extended formulae are A-stable for $k \leq 3$ and stiffly stable for $k \leq 6$. The maximum order of a stiffly stable formula is $p = 9$ (*Cash *[23]).*

*(iii) The SDBDF are A-stable for $k \leq 3$ and stiffly stable for $k \leq 10$. The maximum order of a stiffly stable formula is $p = 11$ (*Hairer, Wanner *[53]).*

(iv) The Skell-Kong*'s formulae are A-stable for $k \leq 3$ and stiffly stable for $k \leq 11$. The maximum order of a stiffly stable formula is $p = 12$ (*Skell-Kong *[53]).*

Table 3.4: SDBDF coefficients for $k \leq 4$

k	α_{k-4}	α_{k-3}	α_{k-2}	α_{k-1}	α_k	β_k	γ_k
1				1	-1	-1	$\frac{1}{2}$
2			$-\frac{1}{4}$	2	$-\frac{7}{4}$	$-\frac{3}{2}$	$\frac{1}{2}$
3		$\frac{1}{9}$	$-\frac{3}{4}$	3	$-\frac{85}{36}$	$-\frac{11}{6}$	$\frac{1}{2}$
4	$-\frac{1}{16}$	$\frac{4}{9}$	$-\frac{3}{2}$	4	$-\frac{415}{144}$	$-\frac{25}{12}$	$\frac{1}{2}$

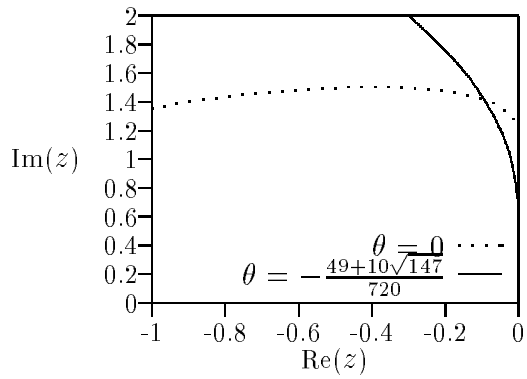

Figure 3.2: Stability domains for SDBDF

Table 3.5: Stability angles of the SDBDF

k	1	2	3	4	5	6	7	8	9	10
p	2	3	4	5	6	7	8	9	10	11
α	90	90	90	89.36	86.35	80.82	72.53	60.71	43.39	12.34

Table 3.6: Stability angles for the SKELL-KONG's formulae

k	1	2	3	4	5	6	7	8	9	10	11
p	2	3	4	5	6	7	8	9	10	11	12
$\gamma_{optim}^{(k)}$	0	0.125	0.122	0.128	0.108	0.096	0.087	0.081	0.076	0.072	0.068
α	90	90	90	89.42	86.97	82.94	77.43	70.22	60.68	47.63	28.68

3.2 Multiderivative linear multistep formulae

3.2.1 Formula

A linear formula with k steps and m-derivatives is an iterative process of the form

$$\boxed{\sum_{i=0}^{m}\sum_{j=0}^{k}(-1)^i\alpha_{ij}h^i y_{n-k+j}^{(i)} = 0, \quad n \geq k,} \tag{3.7}$$

where $\sum_{i=0}^{m}\alpha_{i0}^2\sum_{i=0}^{m}\alpha_{ik}^2 \neq 0$. Note $l_j = \max\{i|\,\alpha_{ij} \neq 0\}$.

3.2.2 Particular cases

- We get the standard linear multistep formulae when $m = 1$ and the second derivative formulae when $m = 2$.
- The onestep methods for $k = 1$ can be rewritten in the form

$$y_{n+1} = y_n + \sum_{i=1}^{m}\frac{h^i}{i!}\left(\alpha_i y_{n+1}^{(i)} + \beta_i y_n^{(i)}\right). \tag{3.8}$$

For example,

1. the TAYLOR's series correspond to $\alpha_i = 0,\ i = 1,\ldots,m$,
2. the formulae

$$y_n = y_{n-1} + h\sum_{k=0}^{p-1}(-h)^k[P^{(p-k-1)}(1)f^{(k)}(y_n) - P^{(p-k-1)}(0)f^{(k)}(y_{n-1})]$$

where

$$P(x) = \frac{1}{(2n)!}\frac{d^n}{dx^n}(x^n(1-x)^n)$$

are A-stable (the PADÉ's approximations to the exponential function [59]).

- The ADAMS type methods are the formula with $(\alpha_{0i}) = (1, 0, ..., 0, -1)$. The ENRIGHT's formulae are an exemple of such methods.
- The BROWN's formulae [12] generalize the backward differentiation formulae. The general formula is

$$\sum_{i=0}^{k} \alpha_i y_{n+i-k} = \sum_{j=1}^{m} h^j \beta_j f_n^{(i-1)}, \quad \sum_{i=0}^{k} |\alpha_i| > 0, \quad \sum_{j=1}^{m} |\beta_j| > 0, \tag{3.9}$$

with $\beta_0 = -\alpha_k,\ (-1)^{j+1}\beta_j \geq 0,\ j = 0, \ldots, m,\ \sum_{i=0}^{k} \alpha_i = 0,\ \alpha_k \neq 0,\ \beta_k \neq 0.$

3.2.3 Order

A generalization of the *barrier order* is due to GENIN [48]: the maximum order of a k-step m-derivative formula is $(m+1)(k+1) - 2$.

The BROWN's methods have the maximum order $p = k + m - 1$ for

$$\alpha_i = (-1)^{k-1} C_k^i (k-1)^{-m}, \quad \beta_j = \frac{(-1)^j}{j!} \sum_{i=0}^{k-1} (-1)^{k-i} C_k^i (k-i)^{j-m},$$

where $i = 0, \ldots, k-1,\ j = 0, \ldots, m.$

3.2.4 Stability

The order of an A-stable formula can be improved introducing the multiderivatives. In Table 3.7 we mention the order of such methods, function on the parameters k and m [68]. With * we have indicated a formula which is only A(α)-stable with $\alpha < 90°$.

Table 3.7: The order of the A-stable k-step m-derivative formulae

$m\backslash k$	1	2	3	4	5	6	7	8
1	1	2						
2		3	4	5*				
3				6	7*			
4					8	9*	10*	11

The DAHLQUIST's second barrier is generalized for the case of a multiderivative formula by the DANIEL-MOORE's conjuncture.

Theorem 3.1. *(DANIEL-MOORE's conjuncture [53]) The maximum order of an A-stable linear multistep formula with m derivatives is $2m$.*

The A-stability algebraic conditions have been studied by JELTSCH. Let

$$\rho_i(\xi) = \sum_{j=0}^{k} \alpha_{ij}\xi^j, \quad r_i(z) = \sum_{j=0}^{k} a_{ij}z^j = (z-1)^k \rho_i\left(\frac{z+1}{z-1}\right), \ i = 0, \ldots, m,$$

$$Q_{ij}(\xi,\mu) = \sum_{s=\max(j,i-m)}^{\min(k,i)} \frac{1}{s!} C_s^j \xi^s \mu^{s-j} \rho_{l+s-i}^{(s)}(\xi), \qquad H(p,q) = \sum_{i=0}^{m}\sum_{j=0}^{k} a_{ij}p^{k-j}q^i.$$

Proposition 3.1.

*(i) A multiderivative multistep formulae is A-stable iff (*JELTSCH *[63])*

1. *$H(p,1)$ is an strictly* HURWITZ *polynomial, i.e it has all the roots in the set $Re\, p < 0$;*
2. *$H(v,p)$ is a* HURWITZ *polynomial, i.e. it has all the roots in the set $Re\, p \leq 0$, for any $v \in \mathbf{R}$;*

*(ii) A multiderivative multistep formulae is A-stable iff (*JELTSCH *[65])*

1. *r_i are real nonzero* HURWITZ *polynomial;*
2. *$r_{i-1}(p)/r_i(p), i = 1, \ldots, m$, are real positive functions for $p \geq 0$;*
3. *all the coefficients a_{ij} have the same sign;*

*(iii) A multiderivative multistep formulae is stiffly stable iff (*JELTSCH *[64])*

1. *it is A_0-stable;*
2. *all the roots of the polynomial $\rho(\xi_0)/(\xi - 1)$ are inside the unitary complex circle;*
3. *if m_j are the multiplicities of the roots of the polynomial $\rho_j(\xi)$ with $|\xi| = 1$, then $m_{l-j} = m_l - j, \ j = 0, \ldots, m_l$;*
4. *if $\overline{\xi}$ is the root of $\rho_l(\xi)$ with $|\overline{\xi}| = 1$, then all the roots of the polynomial $Q_{m_l 0}(\overline{\xi},\mu)$ are positive real values, and if $\overline{\mu}$ is the root of the polynomial $Q_{m_l 0}(\overline{\xi},\mu)$ of k multiplicity, then*

$$Q_{ij}(\overline{\xi},\overline{\mu}) = 0, \qquad \forall i,j \, : \, m_l < i < -j + k + m_l, \ j = 0, \ldots, k-2.$$

Application. Let

$$\rho_0(\xi) = \xi^2 - \xi, \quad \rho_1(\xi) = -\frac{1}{2}(\xi^2 + 1), \quad \rho_2(\xi) = \frac{1}{16}(19\xi^2 - 2\xi - 17),$$

$$\rho_3(\xi) = -\xi^2 - \xi, \quad \rho_4(\xi) = \xi^2 + 2\xi + 1.$$

Applying *(iii)* we can find that the formula is stiffly stable.

Lemma 3.3.

(i) The BROWN*'s formulae are A_0-stable and stiffly-stable for $k \leq \frac{3}{2}(m+1)$ (*JELTSCH *[62]),*

*(ii) If the coefficients of the m derivative formula are independent from the stepsize h and $\alpha_{0i} \leq 0, |\alpha_{iq}| \leq \alpha_{kq}, q = 1, \ldots, m$, then the formula is $A(\pi/(2m))$-stable (*LI *[78]).*

3.3 Block formulae

The building idea is to produce a set of approximations in different interval points at each integration step. The predictor-corrector schemes are the first examples of such composed methods. A RUNGE-KUTTA's process can be also treated as a block methods, but we can evaluate the exact solution in some extra-division points.

These formulae are also referred to as composed methods.

The block methods have some strange properties: when we make a combination of some unstable methods we can get a stable block scheme or an greater order than those of the component formulae. These results was explained by the A-methods theory (ALBRECHT [2]).

The block methods are suitable for parallel implementation.

3.3.1 Formula

The block methods lie in the iterative process

$$\boxed{\sum_{j=-m}^{l-1} \alpha_{ij} y_{kn+j} - h \sum_{j=-m}^{l-1} \beta_{ij} f_{kn+j} = 0, \quad i = 0, \ldots, l-1,} \tag{3.10}$$

where m is the number of starting points, l is the number of points at which we get the solution approximations at one step, and k is the advancing point number with $k \leq l$.

Applying such a method, we follow the steps:

[Step 0]: compute $y_1, \ldots, y_{m-1}$ by a starting procedure;

[Step p]: follow the stages

1. let $x_j = y_{kp+j}$, $j = 0, \ldots, m-1$; solve the algebraic equations on the unknowns $x_m, \ldots, x_{m+l-1}$;
2. store the first output k values as the numerical values of the exact solution y, $y_{kp+j} = x_j$, where $j = m, \ldots, m+k-1$;

3.3.2 Particular cases

- We get the standard linear multistep formulae for $l = 1$.
- The cyclic methods correspond to the case $k = l$. The set of l methods with m starting values is referred as a *l-cyclic m-step method.* The generic form as such formulae for the case $k = m$ is due to BICKART and RUBIN [9]:

$$A_0 Y_n + A_{-1} Y_{n-1} - h(B_0 Y'_n + B_{-1} Y'_{n-1}) = 0 \tag{3.11}$$

  where Y_n is the k-dimensional vector of all aproximations computed at one step, and $A_0,\ B_0,\ A_{-1},\ B_{-1}$ are the $l \times l$ characteristic matrix with $(A_{-1})_{ij} = (B_{-1})_{ij} = 0,\ \forall j \neq 0, i = 0, \ldots, l-1$. For each k, there is at least one k-cyclic k-step method with the order $p = k$ [9].
- The WILLIAMS-HOOQ's formulae generalize the ADAMS methods:

$$y_{nkr} - y_{nk+r-1} = h \sum_{j=0}^{k} \beta_{rj} f_{nk+j}, \qquad 1 \leq r \leq k. \tag{3.12}$$

  A more general formula is given in [123], referred to as *advansed linear multistep formulae*:

$$\sum_{j=0}^{k} \alpha_j y_{n+j-k} = h \sum_{j=n-v_n}^{n+u_n} \beta_{nj} f_{n+j-k}.$$

- The BICKART-PICEL's methods have the following form:

$$\sum_{j=0}^{l} \alpha_{ij} y_{kn+j} - h f_{kn+j} = 0, \quad i = 0, \ldots, l. \tag{3.13}$$

- REINER [116] has studied some block implicit symmetrical formulae, defined on the symmetrical interval $[-1, 1]$ and a symmetrical interval division:

$$y_{n+j} = y_{n-1} + \sum_{k=0}^{l} \beta_{jk} f_{n-1+k}, \quad j = 0, \ldots, l.$$

3.3.3 Similar methods

Introducing the high derivative, we get the WATANABE implicit block multiderivative formulae [122]:

$$y_{n+i} = y_n + h \sum_{j,k} b_{ij} h^k f^{(k)}(t_n + \theta_{ij} h, z(t_n + \theta_{ij} h)), \tag{3.14}$$

where z is the HERMITE's interpolation function of the first $p_i - 1$ derivatives of the exact solution y at t_{n+i}. If Φ_{lm} are the HERMITE's base functions, then:

$$z(t_n + \theta h) = \sum_{l=o}^{s} \sum_{m=0}^{p_l - 1} h^m \Phi_{lm}(\theta) y^{(m)}_{n+l}$$

Note this formula with $[p_0, \ldots, p_s]$. The order of such a formula is at least $\sum_{i=0}^{s} p_i - 1$ (WATANABE [122]).

3.3.4 Stability

The DAHLQUIST's first barrier is generalized by the following result.

Theorem 3.2. *(SLOATE, BICKART [121]) A block method which is zero-stable and stable at infinite has the order*

$$p \leq \begin{cases} m + 2l - 2, & l \geq 2, \\ m + 1, & l = 1. \end{cases}$$

Moreover, if $p \geq 1$ *and* $l = 1$ *then*

$$p \leq \begin{cases} m, & k \text{ is odd}, \\ m + 1, & k \text{ is even}. \end{cases}$$

NEVANLINNA and SIPILA [91] prove the nonexistence of an A-stable block method in which all the approximations are explicitly computed. Moreover, the following theorem is a corollary of the DANIEL-MOORE's conjuncture and a generalization of the DAHLQUIST's second barrier.

Theorem 3.3. *(BICKART, JURY [7]) The order of an A-stable block formula can not exceed* $2l$.

Lemma 3.4.

(i) The BICKART-PICEL*'s cyclic formulae are* $A(\alpha)$*-stable for* $k \leq 10$ *(*BICKART, PICEL *[8]).*

(ii) The WILLIAMS-HOOQ*'s formulae are A-stable for* $1 \leq k \leq 8$ *(*WILLIAMS*[123]).*

(iii) The WATANABE*'s following methods are A-stable:* $[p, p]$ *and* $[p, p + 2]$ *formulae for* $p \geq 1$*, and* $[p, p, p]$ *formulae for* $p \leq 10$ *(*WATANABE *[122]).*

(iv) The REINER*'s formulae have the order* $p \geq 2[(m + 2)/2]$ *and are A-stable for* $l \leq 5$.

For example, in Table 3.8 we mention the stability angle of the BICKART-PICEL formulae.

Table 3.8: Stability angles for BICKART-PICEL's formulae

$p = k$	1	2	3	4	5	6	7	8	9	10
GEAR's BDF	90.0	90.0	86.0	73.4	51.8	17.8	-	-	-	-
BICKART-PICEL	90.0	90.0	88.9	87.7	85.5	82.7	79.5	76.0	72.5	69.6

3.4 Hybrid methods

The linear multistep formula gives some approximate values of the exact solution at some points of the integration interval division. The RUNGE-KUTTA's process produces some approximate values of the exact solution at some points between two consecutive points of the integration interval division. The hybrid methods is a combination since it uses some evaluations at some division points and at some extradivision points.

3.4.1 Formula

A hybrid method with a single extradivision evaluation has the form

$$\boxed{\sum_{j=0}^{k} \alpha_j y_{n+j-k} = h \sum_{j=0}^{k} \beta_j f_{n+j-k} + h\beta_\theta f_{n+\theta},} \tag{3.15}$$

where $f_{n+\theta} = f(t_n + \theta h, y_{n+\theta})$, and $y_{n+\theta}$ is computed by a predictor formula.

3.4.2 Particular cases

- The ENGLAND's methods [37] are implicit predictor-corrector schemes with a hybrid corrector formula and with stability properties identically to the ones of the ENRIGHT's second derivative formulae. The advantage of using these methods is the replacement of the second derivative evaluation with the approximation of the exact solution at an extradivision point. The general formula is the folowing:

$$(P) \qquad y_{n+\theta} = \sum_{j=0}^{k} \alpha_j y_{n+j-k} + h\alpha f(y_n),$$

$$(C) \qquad y_n = y_{n-1} + h \sum_{j=0}^{k} \beta_j f_{n+j-k} + h\beta f(y_{n+\theta}).$$

  The predictor formula is referred to as the interpolation formula, and the corrector formula, as the quadrature formula.

- The BEAUDET's multidivision multistep formulae [5] are defined by the iterative process:

$$y_{n+1} = \sum_{j=0}^{m-1} \alpha_j y_{n-j} + h \sum_{j=0}^{m-1} \sum_{k=1}^{n} \beta_{kj} f_{n-j+\theta_k}, \tag{3.16}$$

  where m is the stepnumber and n is the number of the used derivatives.

- The PATRICIO's second derivative hybrid method [95] is the iterative process

  $$(P) \qquad y_{n+\theta} + \overline{\alpha}_0 y_n = h(\overline{\beta}_0 y'_n + \overline{\beta}_\theta y'_{n+\theta}) + h^2(\overline{\gamma}_0 f'_n + \gamma_\theta f'_{n+\theta}),$$

  $$(C) \quad y_{n+1} + \alpha_0 y_n = h(\beta_0 y'_n + \beta_\theta y'_{n+\theta} + \beta_1 y'_{n+1}) + h^2(\gamma_0 f'_n + \gamma_\theta f'_{n+\theta} + \gamma_1 f'_{n+1}).$$

  One step lies in three stages: compute $\overline{y}_{n+\theta}$ by (P), evaluate $\overline{y}'_{n+\theta}$ and $\overline{f}'_{n+\theta}$, then compute y_{n+1} by (C).

- The CARROLL's hybrid method [17] consists in the scheme

  $$\alpha_0 y_n + \alpha_1 y_{n+\theta} + \alpha_2 y_{n+1} = h f_{n+1}, \quad y_{n+\theta} = y_n + \gamma h[(1-\mu) f_n + \mu f_{n+\theta}],$$

  where $0 < \theta < 1, 0 < \mu \leq 1$.

3.4.3 Stability

The ENGLAND's hybrid methods have the same stability properties as the ENRIGHT's second derivative formulae since the coefficients of the hybrid formulae are choosed such that the characteristic polynomials of both iterative process to be identically. Moreover, the accuracy order is the same. The request of $k+1$ order for the interpolation formula and the request of $k+2$ order for the quadrature formula implies that the predictor-corrector scheme has the order $k+2$. For example, for $k=1$ and $p=3$ we get the coefficients

$$\alpha(\theta) = \theta(\theta-1), \qquad \beta(\theta) = \frac{-1}{6\theta(\theta-1)},$$

$$\alpha_0(\theta) = (\theta-1)^2,\ \beta_0(\theta) = \frac{3\theta-1}{6\theta},\ \alpha_1(\theta) = -(\theta-2)\theta,\ \beta_1(\theta) = \frac{3\theta-2}{6(\theta-1)}.$$

If θ is choosed such that $\beta_0 = 0$ then

$$\theta = \frac{1}{3},\ \alpha = -\frac{2}{9},\ \alpha_0 = \frac{4}{9},\ \alpha_1 = \frac{5}{9},\ \beta = \frac{3}{4},\ \beta_0 = 0,\ \beta_1 = \frac{1}{4}.$$

The parameter θ can be choosed also from the condition that the linear system matrix $1 - h(\beta_k + \beta\alpha_k)f' - h^2\alpha\beta f'^2$ produced applying the NEWTON's iterations for solving the implicit equations to be the square of some matrix (LAUTSCH [76]):

$$\alpha\beta = -\left(\frac{\beta_k + \beta\alpha_k}{2}\right)^2.$$

Lemma 3.5.

(i) The ENGLAND *'s hybrid methods are A-stable for $k \leq 2$, stiffly stable for $k \leq 7$ and of order $p = k+2$ (*ENGLAND *[37]).*

(ii) The PATRICIO *'s second derivative method is stable at infinity of $\overline{\gamma}_0\gamma_\theta = \gamma_0\overline{\gamma}_\theta$, $\gamma_0, \gamma_1 \neq 0$ and A-stable if $\theta \geq \frac{3}{2}$. The A-stability coexists with the maxim order $p = 5$ (*PATRICIO *[95]).*

(iii) The CAROLL *'s method of order $p = 2$ is L-stable (*CARROLL *[17]).*

3.5 Exponential fitting

The basic idea of the exponential fitting method lies in building up a family of integration formulae with some free parameters, others than the stepsize h. The restrictions imposed by the stability request do not affect the stepsize, only these free parameters.

This method is very efficient in the transitory stage of the integration.

Definition 3.1. A numerical method is *exponentially fitted* with the order s at a value $h\lambda$ if it exactly integrates, using a stepsize h, each initial value problem with a solution of the form $P(t)e^{\lambda t}$, where P is an arbitrary polynomial with the degree $deg\, P \leq s$.

More precisely, if the approximation error in the component e^z of the exact solution of the scalar test equation is

$$T(z) = R(z) - e^z, \tag{3.17}$$

and, for some value $q = \lambda h$, $T(q) = 0$ holds, then the numerical solution is *exact* in the discret meaning ($y_n = y(t_n), \forall n \geq 1$ for the corresponding test equation) and the formula is *exponentially fitted* at q. The formula is exponentially fitted with the order s at q if q is a zero of the T function with $s + 1$ multiplicity.

The exponential fitting may be accomplished by two techniques:

- LINIGER-EHLE's technique: determine the free parameters of a scheme in the purpose of the exponential fitting at some values (foe example, see CASH [21], JACKSON [61], LINIGER and WILLOUGHBY [80]);
- ISERLES's technique: approximate the exact solution with a number of methods and compute the mean of the approximate values multiplied by some coefficients determinated by the exponential fitting conditions at some values (for example, see CASH [20]).

The LINIGER-EHLE's strategy for selecting the free parameters supposes that there is a group of eigenvalues of the JACOBIAN's matrix near to the real axis. The formulae with a single free parameter are fitted to the mean of these eigenvalues. If there are two distinct groups of eigenvalues, we must use formulae with two free parameters. If the eigenvalues are changing during the integration, the formulae must be refitted. When the formula has many free parameters, one must be fitted in such a manner that the formula to be stable at infinity (exponential fitted at $q = -\infty$).

Example. Let the μ-rule

$$y_n - y_{n-1} = h[(1 - \mu)f_n + \mu f_{n-1}].$$

For any value of μ, the formula is exponentially fitted at zero with the order at least one. When the value μ is choosed so that the formula is exponentially fitted to an arbitrary point q from the complex plane, then

$$\mu(q) = 1/q + 1/(e^q - 1). \tag{3.18}$$

For $\mu = 0$ we get the EULER's implicit rule which is exponentially fitted at zero and at $-\infty$. The trapezoidal rule is exponentially fitted with second order at zero and we

get it for $\mu = 1/2$. The A-stability condition request that $0 \leq \mu \leq 1/2$. This condition is not restrictive since $\mu(q) : (-\infty, 0] \to [0, 1/2]$ is an one-to-one map. The optimal μ is the value for which

$$v = \min_{\mu} \max_{0 \leq q \leq \infty} |R(q) - e^q|$$

This optimal value is due to MIRANKER [86]: $\mu = 0.122$ for $q = -8.19$, and the objective function minimum value is $v = 0.139$. For EULER's implicit rule the value 0.204 is nearest to the optimal value than the value 1 of the trapezoidal rule.

3.6 Extrapolation method

3.6.1 Formula

Let a sequence of pozitive integers $n_1 < n_2 < n_3 < \cdots$ and the sequence $h_1 > h_2 > h_3 > \cdots$ related by $h_i = H/n_i$. Choosing the positive integers may be performed in various ways:

1. the natural sequence: $1, 2, 3, 4, 5, 6, 7, \ldots$;
2. the powers of the value two: $1, 2, 4, 8, 16, 32, \ldots$;
3. the powers 2^k alternating with 1.5×2^k: $1, 2, 3, 4, 6, 8, 12, 16, 24, 32, \ldots$
4. the harmonic sequence: $2, 6, 10, 14, 22, 34, 50, \ldots$.

Let a integration step-by-step method of p order. We evaluate, for each $i \geq 1$, the numerical solution of the initial value problem on the interval with the length H, computing n_i values at some points of an equidistant division with the stepsize h_i. Note

$$T_{i,0} = y_{n_i}.$$

Let the interpolation polynomial

$$P(h) = e_0 + e_p h^p + e_{p+1} h^{p+1} + \cdots + e_{p+k-1} h^{p+k-1}, \; P(h_i) = T_{i,0}, \; i = j-k, \ldots, j$$

We extrapolate the result by evaluating the limit value at $h \to 0$: $P(0) = e_0 = T_{j,k}$.

The extrapolation table has the following form:

$$\begin{array}{ccc} T_{j,0} & & \\ & T_{j-1,1} & \\ \vdots & \vdots & \cdots \; T_{j,k}. \\ & T_{j-k+1,1} & \\ T_{j-k,0} & & \end{array} \tag{3.19}$$

The extrapolation method improves the initial approximations with each new column of the above table.

Proposition 3.2. *[53] The values $T_{j,k}$ are the results of a numerical method of $p + k$ order.*

3.6.2 Stability

The extrapolation of an A-stable method is not always an A-stable method. We mention the following counterexamples:

- the extrapolation of the trapezoidal rule with $p = 1$ and the natural sequence of positive integers leads to a non-A-stable method [53]. Moreover, when

$$T_{j,0} = y_{h_j}(H), \quad T_{j,k} = T_{j,k-1} + \frac{T_{j,k-1} - T_{j-1,k-1}}{(n_j/n_{j-k})^2 - 1}, \qquad k \geq 1, \tag{3.20}$$

  the resulting method of third order is not stable at infinity.

- CASH [19] proves that the extrapolation scheme based on the sequence $\{n_i\} = \{2, 3, 4, 6, 8, 12, 16, 24, 32, 48, \ldots\}$, the AITKEN-NEVILLE's iterative formulae

$$T_{j,0} = y_{h_j}(H), \quad T_{j,k} = T_{j,k-1} + \frac{T_{j,k-1} - T_{j-1,k-1}}{(n_j/n_{j-k}) - 1}, \qquad k \geq 1$$

  and the EULER's implicit rule, has an nonregular error behaviour in the stiff components.

The A-stability conditions are satisfied when:

1. we use some proper sequences n_j;
2. the approximations $y_{h_j}(H)$ are processed before the extrapolation;
3. we use some special basic methods.

Examples:

- If we use only even numbers in the sequence n_j, an interpolation polynomial with only even powers of the generating variable, the trapezoidal rule and the relationships (3.20), we get an A-stable method (HAIRER, WANNER[53]).

- By smoothing the initial values we can get a method which is stable at infinity. Thus, let

$$T_{j,0} = \frac{1}{4}(y_{n_j-1} + 2y_{n_j} + y_{n_j+1}),$$

  the trapezoidal rule and the natural sequence of positive intergers as stepnumbers. The resulting method is L-stable (LINDBERG [79]).

- Some A-stable methods can be produced using the linear-implicit midlepoint rule. Let $y' = Ay + g(t, y)$ the differential system. The LAWSON's function is $y(t) = e^{At}c(t)$. Transforming the initial system we get a new one in c:

$$c' = e^{-At}g(t, e^{At}c), \qquad g(t, y) = f(t, y) - Ay.$$

  If we apply the midlepoint rule,

$$c_{n+1} = c_{n-1} + 2he^{-At_n}g(t, e^{At_n}c_n),$$

holds, thus,

$$e^{-hA}y_{n+1} = e^{hA}y_n + 2hg(t_n, y_n).$$

We make the approximation $e^{\pm hA} \approx I \pm hA$ and we get the linear-implicit scheme

$$(I - hA)y_1 = y_0 + hg(t_0, y_0),$$

$$(I - hA)y_{k+1} = (I + hA)y_{k-1} + 2hg(t_0 + kh, y_k),\ k = 1, \ldots, n_j$$

The method has the order $p = 2$. Suppose that we use the harmonic sequence. We make a smoothing step,

$$T_{j,0} = (y_{n_j/2+1} + y_{n_j/2-1})/2.$$

and we use the iterative process (3.20). Then $T_{j,0},\ T_{1,1},\ T_{2,1}$ and $T_{2,2}$ are A-stable (BADER, DEUFLHARD [4]).

- The extrapolation scheme based on the EULER linear-implicit rule

$$(I - hJ)(y_{n+1} - y_n) = hf_n, \qquad J \approx f_y(t_n, y_n), \tag{3.21}$$

and the AITKEN-NEVILLE's iterations leads to a method $T_{j,k}$ of the order $k + 1$. For the natural sequence we get A(α)-stability [35].

Chapter 4

NONLINEAR AND MULTIVALUES METHODS

The integration of some nonlinear systems gives us the idea to use some nonlinear interpolation formulae. The first question is concerning to the restrictions imposed in a stiff problem numerical integration. We can construct some explicit nonlinear formulae which are A-stable.

Definition 4.1. A numerical step-by-step method is a *linear* one if applying it to the equation $y' = Ay$, with A an arbitrary matrix, we get a linear equation in the discrete variable y_n.

The classical examples of linear methods are the linear multistep formulae, the predictor-corrector schemes, or the RUNGE-KUTTA's process.

4.1 Linear multistep formulae with matrix coefficients

The real difficulty in integrating a stiff problem is due to the fact that we must often compute some matrix inverses.

4.1.1 Formula

A linear multistep formula with matrix coeficients has the form (LAMBERT [75]):

$$\sum_{j=0}^{k}\left[a_j^{(0)}I+\sum_{i=1}^{s}a_j^{(i)}h^iQ_n^i\right]y_{n+j-k}=h\sum_{j=0}^{k}\left[b_j^{(0)}I+\sum_{i=1}^{s-1}b_j^{(i)}h^iQ_n^i\right]f_{n+j-k}, \qquad (4.1)$$

where Q_n is a variable matrix choosed so that $\|Q(n)\|$ is bounded and the matrix coefficient of y_n is nonsingular. If $b_k^{(j)}=0,\ j=0,\dots,s-1$ the new approximation can be directly computed at each step with an unique matrix inversion. Practically, the $-Q_n$ matrix is an approximation of the JACOBIAN's matrix.

The order of accuracy is independent on Q_n. The method has order p if

$$\sum_{j=0}^{k} a_j^{(i)} = 0, \quad \frac{1}{m!}\sum_{j=0}^{k} j^m a_j^{(i)} = \frac{1}{(m-1)!}\sum_{j=0}^{k} j^{m-1} b_j^{(i)},$$

where $m = 1, \ldots, p-i, \quad i = 0, \ldots, s$.

Some similar methods was proposed by LAMBERT[72] referred to as linear multistep formulae with medium variable coefficients:

$$\sum_{j=0}^{k} \bar{\alpha}_j(t_n) y_{n+j-k} = h \sum_{j=0}^{k} \bar{\beta}_j(t_n) f_{n+j-k}, \tag{4.2}$$

where

$$\bar{\alpha}_j(t_n) = \alpha_j + h a_j(t_n), \quad \bar{\beta}_j(t_n) = \beta_j + h b_j(t_n), \quad |a_j(t)| \le A, \quad |b_j(t)| \le B.$$

for example

$$a_j(t) = a_j^{\circ} q(t), \; b_j(t) = b_j^{\circ} q(t), \; q(t) = -(\partial f/\partial y)(t, y(t)),$$

MAJDA [82] (linear multistep formulae with dynamical filter), BJUREL [11] (modificed linear multistep formulae with exponential coefficients), NORSETT [93] (A-stable generalized ADAMS-BASHFORTH method in which the coefficient of y_{n-1} is e^{hJ}, where $J = \partial f/\partial y$).

4.1.2 Stability

The advantage of these formulae is due to the fact that we can build-up some explicit A-stable formulae. For example, the second oder formula with $s = 1$

$$\left(I + \frac{h}{2}Q_n\right) y_n - [(1+\alpha)I + hQ_n] y_{n-1} + \left(\alpha I + \frac{h}{2}Q_n\right) y_{n-2} =$$

$$= \frac{h}{2}[(3-\alpha) f_{n-1} - (1+\alpha) f_{n-2}], \qquad -1 < \alpha < 1.$$

is A-stable.

The behaviour of the classical linear multistep formulae at the scalar test equation is a prevision model of its behaviour when we integrate a linear system. This fact is not valid also for the formulae with matrix coefficients.

Definition 4.2. A linear multistep formula with variable matrix coefficients is $\overline{\overline{A}}$-*stable* if all the solutions of the difference equation produced applying the method with $Q_n = -A$ to the problem $y' = Ay$, with $\max_i \operatorname{Re}\lambda_i < 0$, $y(t_0) = y_0$, converge to zero when $n \to \infty$, for any stepsize h.

Stability barrier: The maximum order of an $\overline{\overline{A}}$-stable formulae is $2s$.
Example. The method with $s = 1$, $k = 1$, $p = 1$: $a_1^{(0)} = 1$, $a_0^{(0)} = -1$, $b_1^{(0)} = b$, $b_0^{(0)} = 1 - b$, $a_1^{(1)} = a$, $a_0^{(1)} = -a$ is $\overline{\overline{A}}$-stable if $a + b \ge 1/2$. For $b = 0$, we get an explicit formula, and for $b = 1/2, a = 0$, the trapezoidal rule.

4.2 Lambert's nonlinear formulae

Note that the EULER's explicit rule is the result of the interpolation process with

$$I(t) = At + B, \quad y_n = I(t_n), \quad y_{n-1} = I(t_{n-1}), \quad f_{n-1} = I'(t_{n-1})$$

We get the iterative formula by the elimination of the parameters A and B.

4.2.1 Formula

LAMBERT [73] proposes the replacement of the polynomial I with a rational function. For example, we mention some conditions:

$$(L1) \qquad I(t) = \frac{A}{t+B}, \quad y_n = I(t_n),\ y_{n-1} = I(t_{n-1}),\ f_{n-1} = I'(t_{n-1})$$

$$(L2) \qquad I(t) = \frac{At+B}{t+C}, \quad y_{n-i} = I(t_{n-i}),\ i = 0,1,2,\ f_{n-1} = I'(t_{n-1})$$

$$(L3) \qquad I(t) = \frac{At+B}{t+C}, \quad y_{n+i} = I(t_{n-i}),\ i = 0,1,\ f_{n-1} = I'(t_{n-1}),\ f'_{n-1} = I"(t_{n-1})$$

The resulting formulae are the following ones:

$$(L1) \qquad y_n - y_{n-1} = \frac{h y_{n-1} f_{n-1}}{y_{n-1} - h f_{n-1}},$$

$$(L2) \qquad y_n - y_{n-1} = \frac{h(y_{n-1} - y_{n-2}) f_{n-1}}{2(y_{n-1} - y_{n-2}) - h f_{n-1}},$$

$$(L3) \qquad y_n - y_{n-1} = \frac{2h f_{n-1}^2}{2f_{n-1} - h f'_{n-1}},$$

These formulae are applied at each component of the solution. For example, for (L1) and the problem $y' = f(y),\ f = (f^1, \ldots, f^N),\ y(t_0) = y_0$

$$y_n^i - y_{n-1}^i = \frac{h y_{n-1}^i f_{n-1}^i}{y_{n-1}^i - h f_{n-1}^i}, \quad i = 1, \ldots, N, \quad n \geq 1.$$

The first two methods are examples of linear multistep methods with matrix coefficients:

$$(L1) \qquad [I - h\,\mathrm{diag}\,(f_n^i / y_n^i)](y_n - y_{n-1}) = h f_{n-1},$$

$$(L2) \qquad \left[I + \frac{h}{2} Q_n\right] y_n - h Q_n y_{n-1} + \left[-I + \frac{h}{2} Q_n\right] y_{n-1} = 2h f_{n-1},$$

where $-Q_n = \mathrm{diag}\left[2\frac{f_{n-1}^i - (y_{n-1}^i - y_{n-2}^i)/h}{y_{n-1}^i - y_{n-2}^i}\right]$.

4.2.2 Order and stability

A disadvantage of these formulae is the locally unconsistency, i.e. the order of the method is not a constant (it is a variable function on the iterative values). This inconvenient can be avoid applying some linear multistep formula in a neighbourhood of an unconsistent point.

The order p computed by LAMBERT [73] is

$$p_{L1}\begin{cases}\geq 1, y_{n-1}\neq 0\\ =0, \text{altfel},\end{cases} \quad p_{L2}\begin{cases}\geq 2, f_{n-1}\neq 0,\\ =0, \text{altfel},\end{cases} \quad p_{L3}\begin{cases}\geq 2, f_{n-1}\neq 0\\ =1, f_{n-1}=0, f'_{n-1}\neq 0.\end{cases}$$

Lemma 4.1. *[73]*

(i) The formula (L1) is L-stable and its characteristic function is identically to the one of EULER*'s implicit rule.*

(ii) The formulae (L2) and (L3) are A-stable and there characteristic functions are indentically to the one of the trapezoidal rule.

4.3 Exponential methods

Let the CAUCHY's problem associated to the differential system $y' = Ay + g(t, y)$.

A multistep exponential formulae is an iterative process of the form

$$\sum_{i=0}^{k}\alpha_i e^{Ah(k-i)}y_{n+i-k} = h\sum_{i=0}^{k}\Phi_{ki}(Ah)g_{n+i-k}, \quad \alpha_k\neq 0, \tag{4.3}$$

where α_i are some constants independent on h, and $\Phi_{ki}(Ah)$ are dependent on h and A. The formula is a nonlinear one, since the difference equation produced applying the method to a linear system is nonlinear in h.

Practically, the exponentials are numerical approximated, for example by PADÉ's rational function [53].

The advantage of such method is the possibility to use a reasonable stepsize, since the stability condition is $\|\Phi_{kk}(Ah)\|h\|g_y\| < 1$.

Examples:

- The linear-implicit midlepoint rule defined in the extrapolation method section can be a first example.
- For $k = p = 1$ the Φ function are (PREISER, LEE [77])

$$\begin{cases}\Phi_{10}(Ah) = -(Ah)^{-2}(\alpha_0(Ah-I)e^{Ah}-I),\\ \Phi_{11}(Ah) = -(Ah)^{-2}(\alpha_0 e^{Ah}+I+Ah).\end{cases}$$

- The generalized ADAMS-BASHFORTH method, due to CHU [28], has the form

$$y_{n+1} = e^{Ah_{n+1}}y_n + h\sum_{i=0}^{k}\Phi_{ki}y_{n-i+1}, \Phi_{ki} =$$

$$= \int_0^1 e^{A(1-\alpha)h} \prod_{j=1, j\neq i}^{k} \frac{\alpha + j - 1}{j - i} d\alpha, \tag{4.4}$$

For $A = 0$ we get the classical method. If the exponentials are approximated by the PADÉ's function, the method is A-stable.

- In the CERTAIN's method one step consists in two stages

  (P) compute

  $$y_n^P = e^{Ah} y_{n-1} + \int_{t_{n-1}}^{t_n} e^{-A(t-t_n)} g_k^P(t) dt, \tag{4.5}$$

  where $g_k(t)^P$ is the interpolation polynomial of k degree of the $g(t, y(t))$ values at $t_{n-k-1}, t_{n-k}, \ldots, t_{n-1}$. The output is the predictor value;

  (C) using the predictor value, compute

  $$y_n = e^{Ah} y_{n-1} + \int_{t_{n-1}}^{t_n} e^{-A(t-t_n)} g_k(t) dt, \tag{4.6}$$

  starting from the interpolating points $t_{n-k}, t_{n-k+1}, \ldots, t_n$.

If $\deg g \leq k + 1$, then this explicit method is A-stable (MIRANKER [86]).

4.4 Stiff separability

In almost all cases, the eigenvalues of the JACOBIAN's matrix of a nonlinear stiff differential system can be separated in two sets M_1, M_2, with the properties:

(i) $\lambda_j \in M_1$, $\mathrm{Re}\,\lambda_j \ll -1$, $|\mathrm{Im}\,\lambda_j| \leq \rho |\mathrm{Re}\,\lambda_j|$;

(ii) $\lambda_j \in M_2$, $|\lambda_j| \leq \frac{1}{2} c$, with ρ, c of moderate sizes.

Definition 4.3. [10] The system $y' = f(t, y)$ is *stiffly separable*, for a pair (t_0, y_0), if the JACOBIAN's matrix $J = (\partial f / \partial y)(t_0, y_0)$ has the eigenvalues $\{\lambda_i\}$ separable in two sets

$$\min_{1 \leq i \leq s} |\mathrm{Re}\,\lambda_i| \gg \max_{s+1 \leq i \leq N} |\mathrm{Re}\,\lambda_i|$$

The eigenvalues $\lambda_1, \ldots, \lambda_s$ are referred to as the *stiff eigenvalues*. The eigenvectors associated with the stiff eigenvalues determine the *dominant subspace*, and the others, the *dominate subspace*.

Let N the number of system equations and $S = \{1, 2, \ldots, N\}$. If the nonlinear system is stiffly separable, then there are two distinct subsets I and E of S with the properties $I \cup E = S$, $I \cap E =$ and

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix},$$

$$\sigma(J_{11}) = \{\lambda_i | i \in I\}, \quad \sigma(J_{22}) = \{\lambda_e | e \in E\},$$
$$\mathrm{Re}\,(\lambda_i^{(11)}) \ll \mathrm{Re}\,(\lambda_e^{(22)}) < 0, \quad \forall i \in I,\ e \in E$$

The separability supposes that the subsets I and E are not changing the dimension during the integration. This fact excludes a large set of stiff problems.

The purpose of the separability is to reduce the computational effort by applying some explicit formulae to the nonstiff components and some implicit formulae to the stiff ones.

Example. The HOFER's method [57] consists in applying the modified midlepoint rule

$$y_{n+1/2} = y_n + \frac{h}{2} f_n,\ \overline{y}_{n+1} = y_n + h f_{n+1/2},\ y_{n+1} = \frac{1}{2}(y_{n+1/2} + \overline{y}_{n+1} + \frac{h}{2}\overline{f}_{n+1}),$$

to the subset E and the implicit trapezoidal rule to the subset I. The result is the following scheme:

$$y_{n+1/2}^{(E)} = y_n^{(E)} + \frac{h}{2} f_n^{(E)}, \quad y_{n+1/2}^{(I)} = y_n^{(I)} + \frac{h}{4}(f_n^{(I)} + f_{n+1/2}^{(I)}),$$

$$\overline{y}_{n+1}^{(E)} = y_n^{(E)} + h f_{n+1/2}^{(E)}, \quad \overline{y}_{n+1}^{(I)} = y_{n+1/2}^{(I)} + \frac{h}{4}(f_{n+1/2}^{(I)} + \overline{f}_{n+1/2}^{(I)})$$

$$y_{n+1}^{(E)} = \frac{1}{2}(y_{n+1/2}^{(E)} + \overline{y}_{n+1}^{(E)} + \frac{h}{2}\overline{f}_{n+1/2}^{(E)}), \quad y_{n+1}^{(I)} = \overline{y}_{n+1}^{(I)}.$$

An other example is the correction in the dominant subspace, due to ALFELD and LAMBERT [3].

4.5 (A,B)-methods

The (A,B)-methods are also referred to as *general linear methods,* BUTCHER*'s methods*, or *multivalues methods.*

These formula generalizes both the linear multistep formula and the RUNGE-KUTTA's process.

4.5.1 Formula

A general linear formula is an iterative process of the following form

$$\boxed{\begin{aligned} y_i^{(n)} &= \sum_{j=1}^{r} a_{ij}^{(2)} y_j^{(n-1)} + h \sum_{j=1}^{s} b_{ij}^{(2)} f(t_{n-1} + h c_j, Y_j), \quad i = 1, \ldots, r, \\ Y_i &= \sum_{j=1}^{r} a_{ij}^{(1)} y_j^{(n-1)} + h \sum_{j=1}^{s} b_{ij}^{(1)} f(t_{n-1} + h c_j, Y_j), \quad i = 1, \ldots, s, \end{aligned}} \tag{4.7}$$

where $y_i^{(n)}$ are the external stages, and Y_i, the internal stages. As usual, $r = s$ inserting some zeros. Note $u = r + s$.

The matrix equation is

$$y^{(n)} = A_2 y^{(n-1)} + hB_2 f(Y), \qquad Y = A_1 y^{(n-1)} + hB_1 f(Y),$$

or

$$\boxed{Y^{(n)} = AY^{(n-1)} + hBF(Y^{(n)}),}$$

where $Y^{(n)} = (y^{(n)}, Y)$.

We can rewrite the process in a symbolic form

$$\begin{array}{c|cc} c & A_1 & B_1 \\ \hline & A_2 & B_2 \end{array}$$

An equivalent written form is that of an (A, B, C)-method. In the autonomous case, such a method can be described by the difference equation

$$y_i^{(n)} = \sum_{j=1}^{u} a_{ij} y_j^{(n-1)} + h \sum_{j=1}^{u} b_{ij} f(y_j^{(n)}) + h \sum_{j=1}^{u} c_{ij} f(y_j^{(n-1)}). \tag{4.8}$$

The two forms are equivalents. The (A, B)-method is a particular case of a (A, B, C)-method, respectively when $C = 0$. If

$$\overline{y}_i^{(n)} = \begin{cases} y_i^{(n)}, & i \le u \\ y_{i-u}^{(n-1)}, & i > u \end{cases}, \quad \overline{A} = \begin{pmatrix} A & 0 \\ I & 0 \end{pmatrix}, \quad \overline{B} = \begin{pmatrix} B & C \\ 0 & 0 \end{pmatrix},$$

then the $(\overline{A}, \overline{B})$-method produces the same approximations like the (A, B, C)-method.

4.5.2 Particular cases

- The (A,B)-method is explicit when B is an inferior triangular matrix, else the method is implicit.
- The RUNGE-KUTTA's process with q stages is an (A, B) method with $u = q + 1$. We must mention that there are some generalizations of the classical RUNGE-KUTTA's process which are also examples of (A,B)-methods, like the multistep RUNGE-KUTTA methods, or the pseudo RUNGE-KUTTA methods.
- The explicit linear formula with k step,

$$y_n = \sum_{j=1}^{k} \alpha_k y_{n-k} + h \sum_{j=1}^{k} \beta_k f_{n-k},$$

can be write as an (A, B, C)-method with $y_i^{(n)} = y_{n-i+1}$, $u = k$ and

$$A = \begin{pmatrix} \alpha_1 & \dots & \alpha_{k-1} & \alpha_k \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix}, \quad B = 0, \quad C = \begin{pmatrix} \beta_1 & \dots & \beta_k \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix},$$

and as (A,B)-method with $u = k+1$, $y_{k+1}^{(n)} = y_{n-k}$, and

$$A = \begin{pmatrix} \alpha_1 & \dots & \alpha_k & 0 \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & \beta_1 & \dots & \beta_k \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

If the formula is implicit, the first line of B must start with β_0.

- The predictor-corrector schemes can be also written as some (A,B)-methods.
- Other examples of (A,B)-methods are the hybrid methods or the block methods.

4.5.3 Order

The order algebraic conditions was establish by BUTCHER [15].

Definition 4.4. The (A,B,C)-method is *preconsistent* if

$$Ae = e \tag{4.9}$$

and *consistent* if it is preconsistent and, for some vector v (consistency vector)

$$Av + (B+C)e = v + e, \tag{4.10}$$

holds, where $e = (1,\dots,1)^T$.

Applications:

- a linear multistep formula is consistent iff

$$\sum_{j=1}^{k} \alpha_j = 1, \quad \sum_{j=1}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j, \quad v = (-1,-2,\dots,-k-1)^T;$$

- a predictor-corrector scheme is consistent iff the predictor and the corrector formulae are consistent;
- a cyclic scheme is consistent iff all its component formulae are consistent.

4.5.4 Stability

The stability of the (A,B)-method can be algebraic defined.

Definition 4.5.

(i) The matrix A is *stable* if there is a norm produced by a scalar product for which $\|A\| \leq 1$.

(ii) The (A,B)-method is *stable* if the matrix A is stable.

Theorem 4.1. *The (A,B)-method is stable iff the characteristic polynomial of the matrix A satisfies the root condition.*

The stability function is

$$M(z) = A_2 + zB_2(1 - zB_1)^{-1}A_1. \tag{4.11}$$

A general linear method is

(i) A-stable if $\rho(M(z)) \leq 1,\ \forall z \in \mathbf{C}_-$;

(ii) A_0-stable if $\rho(M(x)) \leq 1,\ \forall x \in \mathbf{R}_-$;

(iii) stable at infinity if $\rho(A_2 - B_2B_1^{-1}A_1) \leq 1$;

(iv) zero-stable if $\rho(A_2) \leq 1$.

The DAHLQUIST's second barrier is generalized by the following theorem.

Theorem 4.2. *[14] For an A-stable general linear method, $r \leq s + 1$ holds.*

4.5.5 Convergence

Let $\|\cdot\|$ the euclidian norm on R^N. The discrete values produced by a (A, B)-method are interpolated with an continous function $y_i(h),\ i = 1, \ldots, u$.

Definition 4.6. A general linear method is *convergent* if, for any autonomous CAUCHY's problem with the standard condition for the unicity of the exact solution,

$$\forall t_0, Y_0(h) : \|y_i^{(0)}(h) - y(t_0)\| \underset{h\to 0}{\longrightarrow} 0 \ \Rightarrow \quad \left\| y_i^{(n)}\left(\frac{t - t_0}{n}\right) - y(t)\right\| \underset{n\to\infty}{\longrightarrow} 0, \tag{4.12}$$

holds for $i = 1, \ldots, u$.

Let d a real value, p a vector, and the norms

$$\|w\|_d = \max_{1\leq i\leq s} d^{p_i}|w_i|, \quad d \geq 1, \quad \|A\|_d = \sup_{\|w\|_d\leq 1} \|Aw\|_d = \max_{1\leq i\leq s} \sum_{j=1}^{N} |a_{ij}|d^{p_i - p_j}$$

Definition 4.7. An (A,B)-method has p *order of*

(i) *convergence* if $\forall k' > 0,\ \exists K, h'$ so that:

$$\|\overline{Y}_h^{(0)} - Y_h^{(0)}\|_{1/h} < k' \Rightarrow \|\overline{Y}_h^{(n)} - Y_h^{(n)}\|_{1/h} < K,\ nh \leq T, \forall h' \leq h$$

(ii) *consistency* if there are $\omega, k \in \mathbf{N}$ so that

$$\|\overline{Y}_h^{(n)} - Y_h^{(n)}\|_{1/h} < k,\ \|A^\omega(\overline{Y}_h^{(n)} - Y_h^{(n)})\|_{1/h} < \frac{k}{d},\ nh \leq T, \forall h' \leq h;$$

(iii) *stability* if there is an $\alpha > 1$ such that for any d and $u \geq 1$

$$\|A^u\|_{1/h} < \alpha, \quad \|B^uA\|_{1/h} < \frac{1}{h^{u-1}}\alpha^u, \quad \||B|^u|A|\|_{1/h} < \frac{1}{h^u}\alpha^u.$$

Theorem 4.3. *[31]*

(i) An (A,B)-method is convergent iff it is consistent and stable.

(ii) An (A,B)-method has p order of convergence if it has p order of consistency and p order of stability.

4.6 Multiderivative multivalue method

For the autonomous CAUCHY's problem, the formula is [54]

$$\boxed{\begin{aligned} y_i^{(n)} &= \sum_{j=1}^{r} a_{ij}^{(2)} y_j^{(n-1)} + \sum_{k=1}^{m} h^k \sum_{j=1}^{s} a_{ij}^{(k,2)} f^{(k-1)}(Y_j), \quad i = 1, \ldots, r, \\ Y_i &= \sum_{j=1}^{r} a_{ij}^{(1)} y_j^{(n-1)} + \sum_{k=1}^{m} h^k \sum_{j=1}^{s} a_{ij}^{(k,1)} f^{(k-1)}(Y_j), \qquad i = 1, \ldots, s \end{aligned}} \tag{4.13}$$

and the matrix formula

$$\boxed{Y^{(n)} = AY^{(n-1)} + \textstyle\sum_{i=1}^{m} \frac{h^i}{i!} A^{(i)} (D_y^i (Y^{(n)}))} \tag{4.14}$$

where $A,\ A^{(1)}, \ldots,\ A^{(r)}$ are squared $q+1$-dimensional fixed matrix, characteristic for a method, $q = r + s$ is the stagenumber, m is the maximum derivative degree, and $Y^{(n)} = (y^{(n)}, Y)$.

Particular cases:

- For $m = 1$ we get the (A,B)-methods.
- The explicit methods are characterized by some $A^{(i)}$ inferior triangular matrix with $i = 1, \ldots, m$.
- The multiderivative multistep linear formulae are the particular cases when

$$A = \begin{pmatrix} 0 & 1 & 0 & & \ldots & 0 \\ & 0 & 1 & 0 & \ldots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ & & & & & 1 \\ 0 & -a_1 & & \ldots & & -a_m \end{pmatrix}, \quad A^{(r)} = \begin{pmatrix} 0 & \ldots & 0 \\ \vdots & & \vdots \\ 0 & \ldots & 0 \\ b_1^{(r)} & \ldots & b_{m+1}^{(r)} \end{pmatrix}.$$

- The multiderivative RUNGE-KUTTA's methods are some generalizations of the standard RUNGE-KUTTA process; in this case

$$A = \begin{pmatrix} 0 & \ldots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 1 \end{pmatrix}, \quad A^{(r)} = \begin{pmatrix} a_{11}^{(r)} & \ldots & a_{1q}^{(r)} & 0 \\ \vdots & & \vdots & \vdots \\ a_{q1}^{(r)} & \ldots & a_{qq}^{(r)} & 0 \end{pmatrix}$$

4.7 A-method

The class of A-methods include all the above mentionated formulae and schemes, both linear and nonlinear methods.

The matrix formula is [2]

$$\boxed{Y_{n+1} = AY_n + h\Phi(t_n, Y_n, h), \quad n \geq 0, \quad Y_0 = \Phi(h)} \tag{4.15}$$

where Φ is the starting procedure, and A is a matrix which is independently on the solved problem.

An A-method consist in

1. a starting procedure,
2. the advance formula,
3. a correct validation $Z(t_n, h)$ of the approximate values Y_n: the global error is the difference $Y_n - Z(t_n, h)$.

Examples:

- The linear multistep formula

$$\sum_{j=0}^{k} \alpha_j y_{n+j-k} - h \sum_{j=0}^{k} \beta_j f_{n+j-k} = 0, \qquad \alpha_k = 1$$

can be write as an A-method with $Y_n = (y_n, \ldots, y_{n-k})$, $A = S \odot I$, $\Phi = (e_1 \odot I)\Psi$, where

$$S = \begin{pmatrix} -\alpha_{k-1} & -\alpha_{k-2} & \ldots & -\alpha_1 & -\alpha_0 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}, \tag{4.16}$$

$$\Psi(t_n, Y_n, h) = (1, 0, \ldots, 0)^T [\beta_k f(t_n + h, -\sum_{j=1}^{k} Y^{(j)} + h\Phi(t_n, Y_n, h)) +$$

$$+ \sum_{j=1}^{k} \beta_j f(t_n - (j-1)h, Y^{(j)})].$$

- An example of modified linear multistep formula is the following difference equation:

$$\begin{pmatrix} y_n \\ hy'_n \\ y_{n-1} \\ hy'_{n-1} \end{pmatrix} = \begin{pmatrix} 1/2 & 1 & 1/2 & 1/8 \\ 0 & 0 & 0 & 0 \\ 1/2 & 1/3 & 1/2 & 5/24 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{n-1} \\ hy'_{n-1} \\ y_{n-2} \\ hy'_{n-2} \end{pmatrix} +$$

$$+h(3/8, 1, -1/24, 0)^T \theta(t_{n-1}, y_{n-1}; h),$$

where θ is the solution of the equation

$$f\left(t + h, \frac{1}{2}y + y^2 + \frac{1}{2}y^3 + \frac{1}{8}y^4 + \frac{3}{8}h\theta\right) = 0.$$

If $T^{-1}AT = \text{diag}\,\{1, \lambda_2, \ldots, \lambda_l\}$, then [52]

$$A = E + \lambda_2 E_2 + \cdots + \lambda_l E_l + \overline{E},$$

where $E = T\,\text{diag}\,\{I, 0, 0, \ldots\}T^{-1}$, $E_2 = T\,\text{diag}\,\{0, I, 0, \ldots, \}T^{-1}, \ldots$

Let the local error

$$e_{n+1} = Z(t_n + h, h) - AZ(t_n, h) - h\Phi(t_n, Z(t_n, h), h).$$

Definition 4.8.

(i) An A-method has *p order of consistency* if, for a CAUCHY's problem with an f function which is p times differentiable continuous,

$$e_0 = \mathcal{O}(h^p), \quad E(e_0 + e_1 + \cdots + e_n) + e_{n+1} = \mathcal{O}(h^p), \quad 0 \leq nh \leq c,$$

holds.

(ii) An A-method is *stable* if A^n is uniformely bounded for any $n \geq 0$.

Theorem 4.4. *[52] Let an stable A-method for which*

$$e_0 = \gamma_0 + \gamma_1 h + \cdots + \gamma_{p-1} h^{p-1} + \mathcal{O}(h^p),$$

$$e_{n+1} = \gamma_0(t_n) + \gamma_1(t_1)h + \cdots + \gamma_{p-1}(t_n)h^{p-1} + \mathcal{O}(h^{p+1}),$$

holds with $\gamma_p(\cdot)$ differentiable continuous. The method has p order of convergency iff has p order of consistency.

Part II

DEVELOPING NEW METHODS FOR STIFF PROBLEMS

Chapter 5

SPLIT FORMULAE

5.1 Split ADAMS-MOULTON formulae (SAM)

5.1.1 Motivation

The classical ADAMS-MOULTON formulae, through implicit ones, can not be applied in the numerical integration of a stiff problem, since they stability domains are bounded. There are also the most used zero-stable methods for the nonstiff case.

Specially for the integration of the stiff problems, CASH [24] proposes the split linear formulae. The split form of a linear multistep formula is

$$\sum_{i=0}^{k} \alpha_{k-i} y_{n-i} - h \sum_{i=1}^{k} \beta_{k-i} f_{n-i} = h\theta f(t_n, \overline{y}_n) + h(\beta_k - \theta) f(t_n, y_n), \qquad (5.1)$$

where θ is an arbitrary parameter, and the value $\overline{y}_n$ is produced by a predictor formula. The formula class which was "splitted" by CASH lies on the GEAR's backward differentiation formulae. The result of this procedure is the improvement of the stability angle as for the GEAR's formulae. In the numerical implementation, CASH have use one GEAR's formula, and in the case of numerical unstability (due to some eigenvalues of the JACOBIAN's matrix which are not inside the stability domain of the formula), it makes a switch to the corresponding procedure of the splitted formula.

The study of the CASH's idea generates two questions:

Problem 1. *Can we get some split linear multistep formulae for the integration of stiff systems, using some basic methods that are not efficient in the stiff case?*

Problem 2. *Can we get an improvement of the stability angle also for other formulae than the* GEAR*'s ones?*

We will prove in the next section that the split ADAMS-MOULTON formulae possess some stability properties which make them usefull for the integration of the stiff problems. Thus, it is possible to improve the CASH's code. We can use an ADAMS-MOULTON's formula as basic method and in the case of the numerical instability (that means we have detected an stiff problem) we switch it into the corresponding split formula.

5.1.2 Formula

Note the ADAMS-MOULTON's formula with the k steps

$$y_n = y_{n-1} + h\sum_{j=1}^{k} \overline{\gamma}_j \bigtriangledown^j f_n = y_{n-1} + h\sum_{j=1}^{k} \beta_{k-i} f_{n-i},$$

where

$$\overline{\gamma}_j = (-1)^i \int_0^1 C^j_{-s+1} ds.$$

and

$$\bigtriangledown^0 f_j := f_j, \qquad \bigtriangledown^i f_j := \bigtriangledown^{i-1} f_j - \bigtriangledown^{i-1} f_{j-1}.$$

The local error of the k step formula is

$$TE^{AM}_n = \overline{\gamma}_{k+1} h^{k+2} y^{(k+2)}(t_n) + \mathcal{O}(h^{k+3}),$$

The characteristic polynomials are [99]

$$\rho_{AM}(\xi) = \xi^{k-1}(\xi - 1), \qquad \sigma_{AM}(\xi) = \sum_{i=0}^{k} \overline{\gamma}_i \xi^{k-i}(\xi - 1)^i,$$

$$r_{AM}(z) = \left(\frac{z-1}{2}\right)^k \left(\frac{z+1}{z-1}\right)^{k-1} \left(\frac{2}{z-1}\right) = \left(\frac{z+1}{2}\right)^{k-1},$$

$$s_{AM}(z) = \left(\frac{z-1}{2}\right)^k \sum_{i=0}^{k} \overline{\gamma}_i \left(\frac{z+1}{z-1}\right)^{k-i} \left(\frac{2}{z-1}\right)^i = \sum_{i=0}^{k} \overline{\gamma}_i \left(\frac{z+1}{2}\right)^{k-i}.$$

We propose the following predictor-corrector scheme with both implicit formulae:

$$\boxed{\begin{aligned} &\text{Predictor: } \overline{y}_n - y_{n-1} = h\sum_{i=1}^{k} (\beta_{k-i} + \overline{\beta}_{k-i}\theta) f_{n-i} + (\beta_k - \overline{\beta}_k\theta) h \overline{f}_n, \\ &\text{Corrector: } y_n - y_{n-1} = h\sum_{i=1}^{k} \beta_{k-i} f_{n-i} + (\beta_k - \theta) h f_n + \theta h \overline{f}_n, \end{aligned}} \tag{5.2}$$

where $\overline{f}_n = f(t_n, \overline{y}_n)$, and β_{k-i} are the coefficients of the classical ADAMS-MOULTON's formula with k steps.

In this form, the corrector formula is consistent and zero-stable.

5.1.3 Order

An efficient method in the implementation supposes that the both two nonlinear system of implicit equations, resulting from different equations, to have the same form: we get $\overline{\beta}_k = -1$. We use the simplified NEWTON's iterations for solving the implicit

equations (an unique JACOBIAN's matrix inversion at some start value). The matrix which is necessary to be inversed is the same for both difference equations:

$$(I - h(\beta_k - \theta)J_n), \qquad \text{where} \quad J_n = (\partial f/\partial y)(y_n^{(0)})$$

where $y_n^{(0)}$ is the starting value of the iterative process for solving the implicit predictor formula.

The condition of a predictor formula of k order lies in

$$V^T\overline{\beta} = e_1,$$

where $v_{ij} = i^{j-1}$ are the elements of some VANDERMONDE matrix, and $\overline{\beta} = (\overline{\beta}_{k-1}, \ldots, \overline{\beta}_0)^T$, $e_1 = (1, 0, \ldots, 0)^T$. Since V is inversable, the system has an unique solution

$$\overline{\beta}_{k-i} = (-1)^{i+1}C_k^i, \qquad i = 0, \ldots, k.$$

Thus, the *split* ADAMS-MOULTON *formula with k steps* can be described by the difference equations:

$$\boxed{\begin{aligned} &\text{Predictor: } \overline{y}_n - y_{n-1} = h\sum_{i=1}^{k}(\beta_{k-i} - (-1)^i C_k^i\theta)f_{n-i} + (\beta_k - \theta)h\overline{f}_n, \\ &\text{Corrector: } y_n - y_{n-1} = h\sum_{i=1}^{k}\beta_{k-i}f_{n-i} + (\beta_k - \theta)hf_n + \theta h\overline{f}_n. \end{aligned}} \tag{5.3}$$

where β_{k-i} are the coefficients of the ADAMS-MOULTON's formula with k steps.

In [99] we propose the notation SAM for such a scheme.

Proposition 5.1. *The SAM scheme with k step has $k+1$ order.*

Proof. We use the identity

$$\sum_{i=0}^{k}(-1)^i C_k^i i^k = (-1)^k k!.$$

Since the classical ADAMS-MOULTON's formula has the order $k+1$, the local error of the predictor formula is

$$\overline{TE}_n^{SAM} = \theta h^{k+1}y^{(k+1)}(t_n) + \mathcal{O}(h^{k+2}).$$

Then the local error of the scheme is

$$TE_n^{SAM} = \left[\overline{\gamma}_{k+1}y^{(k+2)}(t_n) - \theta^2\frac{\partial f}{\partial y}(t_n, y(t_n))y_n^{(k+1)}\right]h^{k+2} + \mathcal{O}(h^{k+3}).$$

where $\overline{\gamma}_{k+1}$ is the error coefficient of the ADAMS-MOULTON's formula. □

5.1.4 Stability

Characteristic equation

The characteristic equation of the predictor formula is

$$\overline{\rho}_{SAM}(\xi) - q\overline{\sigma}_{SAM}(\xi) = 0, \qquad q = h\lambda,$$

where

$$\begin{gathered}\overline{\rho}_{SAM}(\xi) = \rho_{AM}(\xi) = \xi^{k-1}(\xi - 1),\\ \overline{\sigma}_{SAM}(\xi) = \sigma_{AM}(\xi) - \theta\sum_{i=0}^{k}(-1)^i C_k^i \xi^i = \sigma_{AM}(\xi) - \theta(\xi - 1)^k.\end{gathered} \tag{5.4}$$

For the corrector formula the characteristic equation has the two degree in q:

$$\rho_{SAM}(\xi) - q\sigma_{SAM}(\xi) + q^2\tau_{SAM}(\xi) = 0, \qquad q = h\lambda$$

The characteristic polynomials can be expresssed in terms of ρ_{AM} and σ_{AM}.

Lemma 5.1. *The characteristic polynomials of the split scheme (5.3) are*

$$\begin{gathered}\rho_{SAM}(\xi) = \overline{\rho}_{SAM}(\xi) = \rho_{AM}(\xi) = \xi^{k-1}(\xi - 1),\\ \sigma_{SAM}(\xi) = (\beta_k - 2\theta)\rho_{AM}(\xi) + \sigma_{AM}(\xi),\\ \tau_{SAM}(\xi) = (\beta_k - 2\theta)\rho_{AM}(\xi) + \theta^2(\xi - 1)^k.\end{gathered} \tag{5.5}$$

Proof. Applying the scheme to the scalar test equation $y' = h\lambda$ we get the difference equations

$$[1 - (\beta_k - \theta)q]\overline{y}_n = y_{n-1} + q\sum_{i=1}^{k}(\beta_{k-i} - (-1)^i C_k^i\theta)y_{n-i},$$

$$[1 - (\beta_k - \theta)q]y_n = y_{n-1} + q\sum_{i=1}^{k}\beta_{k-i}y_{n-i} + \theta q\overline{y}_n,$$

where $q = h\lambda$. We multiply the first equation with $1 - (\beta_k\theta)q$ and we replace $\overline{y}_n$ with its value from the first equation. It results an equation in second degree on q. Replacing $y_{n-i} \to \xi^{k-i}$ we get the characteristic equation.□

In terms of the (r, s) polynomials, the characteristic equation of the predictor formula is

$$\overline{r}_{SAM}(z) - q\overline{s}_{SAM}(z) = 0,$$

and that of the split scheme is

$$r_{SAM}(z) - qs_{SAM}(z) + q^2 t_{SAM}(z) = 0,$$

where

$$r_{SAM}(z) = \overline{r}_{SAM}(z) = r_{AM}(z) = \left(\frac{z+1}{2}\right)^{k-1},$$

$$\overline{s}_{SAM}(z) = r_{AM}(z) - \theta,$$

$$s_{SAM}(z) = (\beta_k - 2\theta) r_{AM} + s_{AM}(z),$$

$$t_{SAM}(z) = s_{AM}(z) + \theta^2.$$

Example. For the case $k = 2$, the characteristic equation is

$$az^2 + bz + c = 0,$$

where

$$a \equiv a(q,\theta) := q\left[\left(\frac{5}{12} - 2\theta\right) q - 1\right],$$

$$b \equiv b(q,\theta) := (q-2)\left[\left(\frac{5}{12} - 2\theta\right) q - 1\right],$$

$$c \equiv b(q,\theta) := \left(4\theta^2 + \frac{2}{3}\theta - \frac{5}{36}\right) q^2 + \left(4\theta - \frac{1}{2}\right) q + 2.$$

Studying the characteristic equations, we can prove the A_0-stability of some SAM schemes.

Because the characteristic equation has two degree in q, the boundary of the stability domain is defined by the following two curves:

$$C_1 = \left\{ \frac{\sigma_{SAM}(e^{i\alpha}) + \sqrt{\sigma^2_{SAM}(e^{i\alpha}) - 4\rho_{SAM}(e^{i\alpha})\tau_{SAM}(e^{i\alpha})}}{2\tau_{SAM}(e^{i\alpha})} \mid 0 \le \alpha \le 2\pi \right\},$$

$$C_2 = \left\{ \frac{\sigma_{SAM}(e^{i\alpha}) - \sqrt{\sigma^2_{SAM}(e^{i\alpha}) - 4\rho_{SAM}(e^{i\alpha})\tau_{SAM}(e^{i\alpha})}}{2\tau_{SAM}(e^{i\alpha})} \mid 0 \le \alpha \le 2\pi \right\}$$

A_0-stability

The proof that the split schemes are A_0-stable when $k \le 4$ is not easy to be performed using the general formula.

Case $k = 1$. The onestep corresponding method is based on the trapezoidal rule:

$$(P) \qquad \overline{y}_n - y_{n-1} = \theta h f(t_n, \overline{y}_n) + (1-\theta) h f(t_{n-1}, y_{n-1}),$$

$$(C) \qquad y_n - y_{n-1} = \frac{h}{2} f(t_{n-1}, y_{n-1}) + \theta h f(t_n, y_n) + \left(\frac{1}{2} - \theta\right) h f(t_n, \overline{y}_n). \quad (5.6)$$

CASH [24] proves that this scheme is A-stable iff $\theta \ge 1/4$.

Case $k = 2$. The split scheme with two steps has the following form:

$$(P) \qquad \overline{y}_n - y_{n-1} = h\left[\left(\frac{5}{12} - \theta\right) f(t_n, \overline{y}_n) + \right.$$

$$\left. + \left(\frac{2}{3} + 2\theta\right) f(t_{n-1}, y_{n-1}) - \left(\frac{1}{12} + \theta\right) f(t_{n-2}, y_{n-2})\right],$$

$$(C) \qquad y_n - y_{n-1} =$$

$$= h\left[\left(\frac{5}{12} - \theta\right) f(t_n, y_n) + \frac{2}{3} f(t_{n-1}, y_{n-1}) - \frac{1}{12} f(t_{n-2}, y_{n-2})\right] + \theta h f(t_n, \overline{y}_n), \quad (5.7)$$

where $\overline{y}_n$ is the predicted value, and y_n is the corrected value, the final one.

The predictor formula has order two, and the split scheme, the order $p = 3$. If the system function is sufficient differentiable, then

$$\overline{TE}_n = \theta h^3 y_n^{(3)} + \mathcal{O}(h^4),$$

and the local discretization error of the scheme is

$$TE_n = \left[\frac{1}{24} y_n^{(4)} - \theta^2 \frac{\partial f}{\partial y}(y_n) y_n^{(3)}\right] h^4 + \mathcal{O}(h^5).$$

Lemma 5.2. *[107] If* $\theta \in (-\infty, -(1+\sqrt{6})/12] \cup [(\sqrt{6}-1)/12, 1/8]$, *then the third order split* ADAMS-MOULTON *scheme (5.7), is* A_0*-stable.*

A strongest result was given in [99]: the (5.7) scheme is A_0-stable iff

$$\theta \in \left(-\infty, \frac{-1-\sqrt{6}}{12}\right] \cup \left[\frac{\sqrt{6}-1}{12}, \frac{5}{24}\right]. \quad (5.8)$$

For some θ-values the predictor formula has good stability property. For example, for $\theta = -1/3$, the predictor formula is A-stable, and for $\theta < -1/12$ is A_0-stable.

Case $k = 3$. The split scheme with three steps is

$$(P) \qquad \overline{y}_n - y_{n-1} = h\left[\left(\frac{9}{24} - \theta\right) f(t_n, \overline{y}_n) + \left(\frac{19}{24} + 3\theta\right) f(t_{n-1}, y_{n-1}) - \right.$$

$$\left. - \left(\frac{5}{24} + 3\theta\right) f(t_{n-2}, y_{n-2}) + \left(\frac{1}{24} + \theta\right) f(t_{n-3}, y_{n-3})\right],$$

$$(C) \qquad y_n - y_{n-1} = h\left[\left(\frac{9}{24} - \theta\right) f(t_n, y_n) + \frac{19}{24} f(t_{n-1}, y_{n-1}) - \right.$$

$$\left. - \frac{5}{24} f(t_{n-2}, y_{n-2}) + \frac{1}{24} f(t_{n-3}, y_{n-3})\right] + \theta h f(t_n, \overline{y}_n). \quad (5.9)$$

The predictor formula has third order, and the split scheme, the order $p = 4$. The local discretization error of the predictor formula is

$$\overline{TE}_n = \theta h^4 y_n^{(4)} + \mathcal{O}(h^5),$$

and that of the split scheme

$$TE_n = \left[\frac{19}{720} y_n^{(5)} - \theta^2 \frac{\partial f}{\partial y}(y_n) y_n^{(4)}\right] h^5 + \mathcal{O}(h^6).$$

Lemma 5.3. *[112] If* $\theta \in [-(2+\sqrt{10})/8, -(2+\sqrt{22})/24]$, *then the split* ADAMS-MOULTON *scheme with three steps and of four order of accuracy, (5.9), is* A_0*-stable.*

From the point of view of the stability and of the accuracy, the θ optimal value is $\theta = -(2+\sqrt{22})/24$.

Case $k = 4$. Let the split scheme with four steps

$$(P) \qquad \overline{y}_n - y_{n-1} = h\left[\left(\frac{251}{720} - \theta\right) f(t_n, \overline{y}_n) + \left(\frac{323}{360} + 4\theta\right) f(t_{n-1}, y_{n-1}) - \left(\frac{11}{30} + \right.\right.$$

$$\left.\left. + 6\theta\right) f(t_{n-2}, y_{n-2}) + \left(\frac{53}{360} + 4\theta\right) f(t_{n-3}, y_{n-3}) - \left(\frac{19}{720} + \theta\right) f(t_{n-4}, y_{n-4})\right],$$

$$(C) \qquad y_n - y_{n-1} = h\left[\left(\frac{251}{720} - \theta\right) f(t_n, y_n) + \frac{323}{360} f(t_{n-1}, y_{n-1}) - \right.$$

$$\left. - \frac{11}{30} f(t_{n-2}, y_{n-2}) + \frac{53}{360} f(t_{n-3}, y_{n-3}) - \frac{19}{720} f(t_{n-4}, y_{n-4})\right] + \theta h f(t_n, \overline{y}_n). \qquad (5.10)$$

The predictor formula has four order, and the split scheme, the order $p = 5$. The local discretization error of the predictor formula is

$$\overline{TE}_n = \theta h^5 y_n^{(5)} + \mathcal{O}(h^6),$$

and that of the split scheme is

$$TE_n = \left[\frac{3}{160} y_n^{(6)} - \theta^2 \frac{\partial f}{\partial y}(y_n) y_n^{(5)}\right] h^6 + \mathcal{O}(h^7).$$

Lemma 5.4. *[96] For* $\theta = -(49 + 10\sqrt{147})/720$, *the split* ADAMS-MOULTON *scheme of five order and four steps, (5.10), is* A_0*-stable.*

Case $k \geq 5$. The split schemes are not A_0-stable.

Proposition 5.2. *For each* $k \leq 4$ *there are two values* a_k, $b_k \in \mathbf{R}$, *so that the split* ADAMS-MOULTON *formula with* k *steps and* $k+1$ *order is* A_0*-stable at least for* $\theta \in [a_k, b_k]$.

Proof. Using the last lemmata, the request values are well defined:

$$a_k = \begin{cases} 1/4, & k = 1 \\ -a, & k = 2 \\ -(2+\sqrt{10})/8, & k = 3 \\ -(49+10\sqrt{147})/720, & k = 4 \end{cases} \quad b_k = \begin{cases} a, & k = 1 \\ -(1+\sqrt{6})/12, & k = 2 \\ -(2+\sqrt{22})/24, & k = 3 \\ -(49+10\sqrt{147})/720, & k = 4 \end{cases}$$

where a is an arbitrary real pozitive number.□

Stiff stability

It is posible that the proposed scheme are not only A_0-stable, but also stiffly stable. In this purpose, we study the form of stability domains. We have plotted the two curves which are defining the boundaries of the stability domains.

In Figure 5.1 we have plotted the curves in a neighbourhood of the origin of the complex plane for the case $k = 2$ and some special values of the θ parameter.

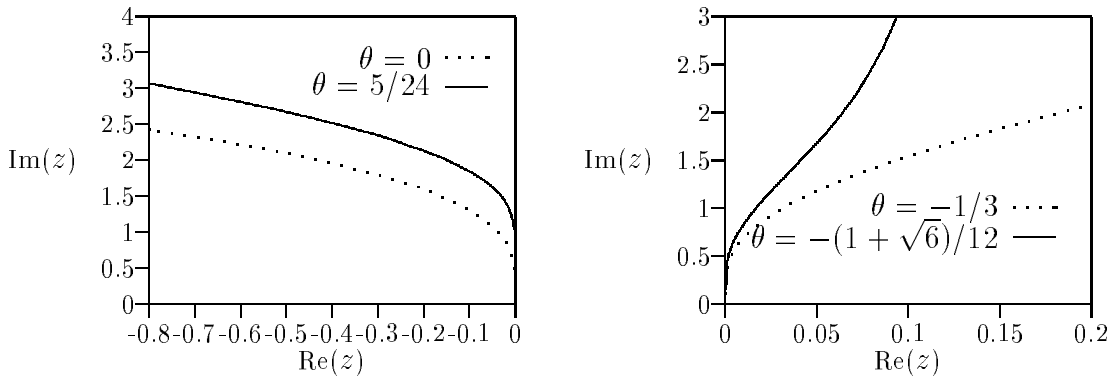

Figure 5.1: Root locus curves in the case $k = 2$ and for some θ values

Remarks:

1. the boundary of the stability domain of the ADAMS-MOULTON's formula with two steps corresponds to the case $\theta = 0$;

2. for some θ negative values, the stability domains includes the set C_-, so the methods are not only stiffly stable, but also A-stable. For example, we have plotted the boundaries for $\theta = -(1 + \sqrt{6})/12$ and $-1/3$. The methods are A-stable at least for $\theta \leq -(1 + \sqrt{6})/12$.

Because of the A-stability property in the condition of an third order of accuracy, the split scheme with two step is an example of breaking the DAHLQUIST's second barrier.

In the case $k = 3$ we get the plots from the Figure 5.2.

For the value $\theta = 0$ the stability domain is inside the region delimited by the plotted curve (the classical method). For the negative θ value, the stability domain includes the negative real axis; the method is not only A_0-stable but it is also stiffly stable.

The constants from the stiff stability definition was computed for some θ values using a numerical method (Table 5.1).

The boundaries of stability domains of the split scheme in the case $k = 4$ and for two θ values are plotted in Figure 5.3. For the negative θ value the domain includes the

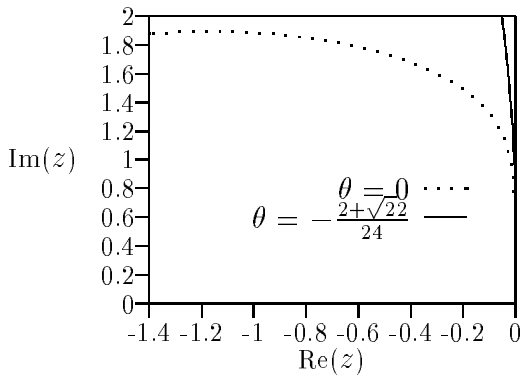

Figure 5.2: Root locus curves for split ADAMS-MOULTON scheme with $k = 3$ and for two θ values

real negative half-axis, i.e. the method is stiffly stable. The stiff stability parameters are $D = 0.034$, $\alpha = 88.260$. The stability domain of the classical method corresponds to $\theta = 0$.

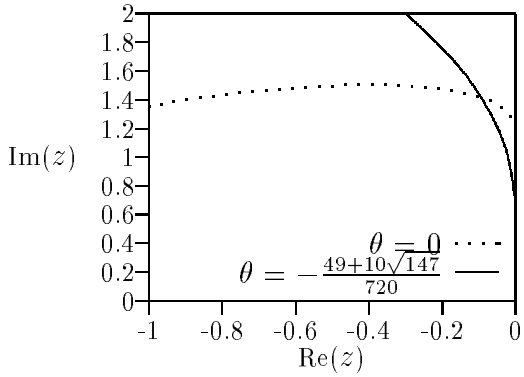

Figure 5.3: Root locus curve for the split scheme with four steps

5.2 Split second derivative multistep method

5.2.1 Motivation

The GEAR's backward differentiation formulae are stiffly stable for $k \leq 6$. Splitting them, we get an improvement of the stability angles. On other hand, the GEAR's formula with $p = k = 3$ is only stiffly stable, while the split scheme is A-stable.

Table 5.1: Stiff stability parameters

θ	D	α
$-(2+\sqrt{22})/24$	0.062	87.052
$-7/24$	0.018	89.296
$-5/8$	0.055	87.360
$-(2+\sqrt{10})/8$	0.021	89.169

CASH[24] suggests a numerical code based on the GEAR's formulae and an algorithm to determine the numerical unstability and the switching procedure to the split schemes.

The CASH's idea can be extended to other formula classes than the backward differentiation methods. We mean some methods with stability properties requested by the integration of a stiff problem. For example, we consider

- the ENRIGHT's second derivative formulae (generalizations of the ADAMS's formulae);
- the second derivative backward differentiation formulae (generalizations of the GEAR's formulae).

5.2.2 Formula

There are three natural ways to generalize the split linear multistep formula using the second derivative:

1. splitting the first derivative, f, and the second derivative, g, of the solution computed in the current point of the division, t_n;
2. splitting only the second derivative of the solution, g, computed in the current point of the division, t_n;
3. splitting only the first derivative of the solution, f, computed in the current point of the division, t_n.

We analyse the results of the splitting method on two formula classes:

- the second derivative multistep methods proposed by ENRIGHT;
- the second derivative backward differentiation methods.

We get some new schemes with stability properties requested by the stiff system only following the first two above mentioned ways. The stability angle can be improved only following the first way.

We propose the name of *second derivative split linear multistep formula* for the following generic equation

$$\boxed{\begin{aligned}&\sum_{i=0}^{k}\alpha_{k-i}y_{n-i}-h\sum_{i=1}^{k}\beta_{k-i}f(t_{n-i},y_{n-i})+h^2\sum_{i=1}^{k}\gamma_{k-i}g(t_{n-i},y_{n-i})=\\&=hb\theta f(t_n,\overline{y}_n)+h(\beta_k-b\theta)f(t_n,y_n)-h^2\theta g(t_n,\overline{y}_n)-h^2(\gamma_k-\theta)g(t_n,y_n).\end{aligned}} \tag{5.11}$$

where $\overline{y}_n$ is a value which is computed by a predictor formula, and $g=y"$.

A particular case of the equation (5.11) corresponds to $b=0$, and it is named *split second derivative linear multistep formula*:

$$\begin{aligned}&\sum_{i=0}^{k}\alpha_{k-i}y_{n-i}-h\sum_{i=0}^{k}\beta_{k-i}f(t_{n-i},y_{n-i})+h^2\sum_{i=1}^{k}\gamma_{k-i}g(t_{n-i},y_{n-i})+\\&+h^2\theta g(t_n,\overline{y}_n)+h^2(\gamma_k-\theta)g(t_n,y_n)=0.\end{aligned} \tag{5.12}$$

with $\overline{y}_n$ produced by a predictor formula.

For the convergence of the formulae (5.11) and (5.12), we use the coefficients α_{k-i}, β_{k-i}, γ_{k-i} of a consistent and zero-stable second derivative formula and a predicted value produced by a consistent formula.

We impose the following two rules for the predictor-corrector scheme:

(i) we choose the predictor formula so that, for the case $\theta=0$, the scheme is equivalent to the nonsplit corrector formula, i.e. the predictor formula has the following form

$$\begin{aligned}&\alpha_k\overline{y}_n-h(\beta_k+\overline{\beta}_k\theta)f(t_n\overline{y}_n)+h^2(\gamma_k-\theta)g(t_n,\overline{y}_n)+\sum_{i=1}^{k}(\alpha_{k-i}+\overline{\alpha}_{k-i}\theta)y_{n-i}-\\&-h\sum_{i=1}^{k}(\beta_{k-i}+\overline{\beta}_{k-i}\theta)f(t_{n-i},y_{n-i})+h^2\sum_{i=1}^{k}(\gamma_{k-i}+\theta\overline{\gamma}_{k-i})g(t_{n-i},y_{n-i})=0;\end{aligned} \tag{5.13}$$

(ii) solving the implicit equations, we use a simplified NEWTON procedure with an unique matrix inversion and we suppose that the inversed matrix is the same for both predictor and corrector implicit difference equation, i.e.

$$\overline{\beta}_k=-b.$$

If $y_n^{(0)}$ is a starting value for the predictor formula, an initial approximation at t_n, and we solve an autonomous problem, then the inversed matrix is

$$I-h(\beta_k-b\theta)\frac{\partial f}{\partial y}(y_n^{(0)})+h^2\frac{\partial}{\partial y}\frac{df}{dt}(y_n^{(0)})=I-h(\beta_k-b\theta)J_n+h^2(\gamma_k-\theta)J_n^2$$

where $J=\partial f/\partial y(y_n^{(0)})$.

We resume us to the study of the order and stability properties of the schemes for which $\gamma_{k-1} = \cdots = \gamma_0 = 0$ and

1. $\alpha_k = -\alpha_{k-1} = 1,\ \alpha_{k-2} = \cdots = \alpha_0 = 0,\ \beta_k \neq 0;$
2. $\alpha_k \neq 0,\ \beta_k = 1, \qquad \beta_{k-1} = \cdots = \beta_0 = 0.$

Note that there are two cases: $b \neq 0$ and $b = 0$.

5.2.3 Formula with both split derivatives. Split ENRIGHT schemes (SES)

Using as basic method an ENRIGHT's second derivative formula or a second derivative backward differentiation formula, it is possible to improve the stability angle by the splitting procedure.

Formula

The proposed scheme, referred to as the *split* ENRIGHT *scheme*, has the following form:

$$\boxed{\begin{aligned}(P)\quad \overline{y}_n &= y_{n-1} + h\sum_{i=1}^{k}(\beta_{k-i} + \overline{\beta}_{k-i}\theta)f(t_{n-i}, y_{n-i}) + h(\beta_k - b\theta)f(t_n, \overline{y}_n) - \\ &\quad -h^2(\gamma_k - \theta)g(t_n, \overline{y}_n), \\ (C)\quad y_n &= y_{n-1} + h\sum_{i=1}^{k}\beta_{k-i}f(t_{n-i}, y_{n-i}) + h(\beta_k - b\theta)f(t_n, y_n) \\ &\quad +hb\theta f(t_n, \overline{y}_n) - h^2(\gamma_k - \theta)g(t_n, y_n) - h^2\theta g(t_n, \overline{y}_n).\end{aligned}} \tag{5.14}$$

where $\beta_{k-i},\ \gamma_k$ are the ENRIGHT's method coefficients, and $\overline{\beta}_{k-i}$ are computed from the maximum order conditions for the predictor formula. Note SES_k the above scheme.

Order

The ENRIGHT's classical formula with k steps is of $k+2$ order. Note c_{k+3}, the error constant of this formula.

Suppose that the predictor formula, in the given form, has the maximum order of accuracy, $p = k + 1$. From the order algebraic condition we get

$$b = -\sum_{i=1}^{k}\overline{\beta}_{k-i}$$

and the system

$$V^T\overline{\beta} = e_1,$$

where $v_{ij} = i^j$ are the elements of a VANDERMONDE matrix, and $\overline{\beta} = (\overline{\beta}_{k-1}, \ldots, \overline{\beta}_0)^T$, $e_1 = (1, 0, \ldots, 0)^T$. Since V is invertible, the system is uniquely solvable and

$$\overline{\beta}_{k-i} = (-1)^i C_k^i / i, \qquad i = 1, \ldots, k. \tag{5.15}$$

By induction method, one can find that

$$b = -\sum_{i=1}^{k} \frac{1}{i}. \tag{5.16}$$

Proposition 5.3. *The split* ENRIGHT *scheme with k steps has $k+2$ order.*

Proof. In the conditions (5.15) and (5.16), the predictor formula has the local truncation error

$$y(t_n) - \overline{y}_n = \overline{TE}_n^{SES} = \frac{1}{k+1}\theta h^{k+2} y^{(k+2)}(t_n) + \mathcal{O}(h^{k+3}),$$

so that, the SES scheme has the following local truncation error:

$$TE_n^{SES} = \left(c_{k+3} y^{(k+3)}(t_n) + \theta^2 \frac{d}{k+1} \frac{\partial f}{\partial y}(t_n, y(t_n)) y^{(k+2)}(t_n) \right) h^{k+3} + \mathcal{O}(h^{k+4}),$$

and its order is $k+2$, the same like that of the classical ENRIGHT's formula with k steps.□

Stability

Since the ENRIGHT's methods are A-stable for $k = 1, 2$ (it is not necessary to improve the stability angle) our atention is concetrated on the cases when $k \geq 3$. Firstly, we study the possibility to obtain A-stable schemes, for $k = 3$.

Case $k = 3$. The corresponding scheme is the following

$$(P) \quad \overline{y}_n = y_{n-1} + h\left[\left(\frac{307}{540} + \frac{11}{6}\theta \right) f(t_n, \overline{y}_n) + \left(\frac{19}{40} - 3\theta \right) f(t_{n-1}, y_{n-1}) + \left(-\frac{1}{20} + \right.\right.$$

$$\left.\left. + \frac{3}{2}\theta \right) f(t_{n-2}, y_{n-2}) + \left(\frac{7}{1080} - \frac{1}{3}\theta \right) f(t_{n-3}, y_{n-3}) \right] - \left(\frac{19}{180} + \theta \right) h^2 g(t_n, \overline{y}_n),$$

$$(C) \quad y_n = y_{n-1} + h\left[\left(\frac{307}{540} + \frac{11}{6}\theta \right) f(t_n, y_n) + \frac{19}{40} f(t_{n-1}, y_{n-1}) - \frac{1}{20} f(t_{n-2}, y_{n-2}) + \right.$$

$$\left. + \frac{7}{1080} f(t_{n-3}, y_{n-3}) \right] - \frac{11}{6}\theta h f(t_n \overline{y}_n) - \left(\frac{19}{180} + \theta \right) h^2 g(t_n, y_n) + \theta h^2 g(t_n, \overline{y}_n). \tag{5.17}$$

The local truncation error has the following form

$$TE_n^{SDE_3} = \left(\frac{3}{800} y_n^{(6)} - \frac{11}{24}\theta^2 \frac{\partial f}{\partial y}(y_n) y_n^{(5)} \right) h^6 + \mathcal{O}(h^7).$$

For the integration of a stiff system, we are asking stiff stability for the scheme (5.17). A necessary condition for this kind of stability is the A_0-stability.

Proposition 5.4. *[109] For $\theta \geq 0$, the split* ENRIGHT *scheme (5.17) is A_0-stable.*

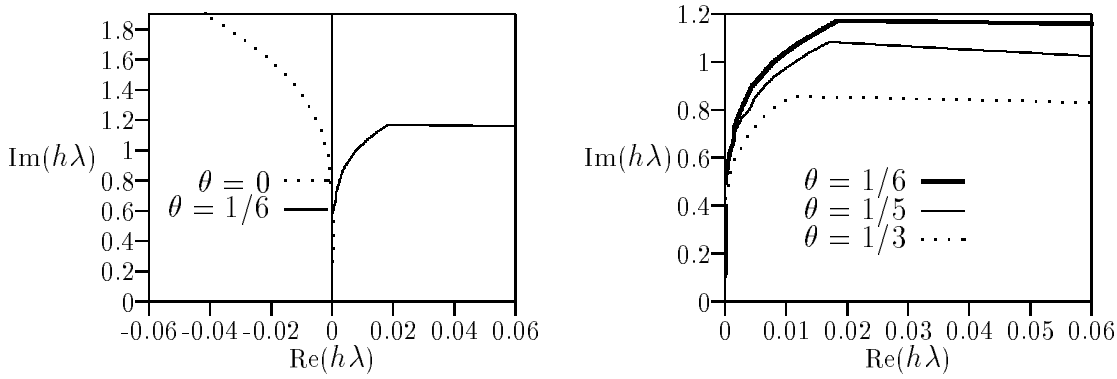

Figure 5.4: Root locus curves for split ENRIGHT scheme with $k = 3$ and $b \neq 0$

We have plotted in Figure 5.4 the root locus curves of the split scheme with $k = 3$. The curves are symmetrically to the real axis. To find stiff stability characteristics we analyse some domain surrounding the origin of the complex plane. For each specific θ, the stability domains include the half-planes defined by the plotted locus curves and the real negative half-axis.

The plots show that, for some θ values which satisfy the inequality $\theta \geq 1/6$, the split scheme is not only stiffly stable, but also A-stable and has a stability domain greatest than the one of the associated classical method. The A-stability property was tested also for smallest values than the above mentioned θ. For instance, for $\theta = 1/12$, the scheme is A-stable and has a resonable error constant which is dominated by c_{k+3}.

Note that the classical method corresponds to $\theta = 0$ and it is only stiffly stable. Thus the A-stable schemes break the DAHLQUIST's barrier.

Case $k = 4$. The corresponding split scheme consists in the following equations:

$$\overline{y}_n = y_{n-1} + h\left[\left(\frac{3133}{5760} + \frac{25}{12}\theta\right) f(t_n, \overline{y}_n) + \left(\frac{47}{90} - 4\theta\right) f(t_{n-1}, y_{n-1}) + \left(-\frac{41}{480} +\right.\right.$$

$$(P) \qquad \left. +3\theta\right) f(t_{n-2}, y_{n-2}) + \left(\frac{1}{45} - \frac{4}{3}\theta\right) f(t_{n-3}, y_{n-3}) + \left(-\frac{17}{5760} +\right.$$

$$\left.\left. +\frac{1}{4}\theta\right) f(t_{n-4}, y_{n-4})\right] - \left(\frac{13}{32} + \theta\right) h^2 g(t_n, \overline{y}_n),$$

$$y_n = y_{n-1} + h\left[\left(\frac{3133}{5760} + \frac{25}{12}\theta\right) f(t_n, y_n) + \frac{47}{90} f(t_{n-1}, y_{n-1}) - \frac{41}{480} f(t_{n-2}, y_{n-2}) +\right.$$

$$(C) \qquad \left. +\frac{1}{45} f(t_{n-3}, y_{n-3}) - \frac{17}{5760} f(t_{n-4}, y_{n-4})\right] - \frac{25}{12}\theta h f(t_n, \overline{y}_n) -$$

$$-\left(\frac{13}{32}+\theta\right)h^2 g(t_n,y_n)+\theta h^2 g(t_n,\overline{y}_n). \tag{5.18}$$

The local truncation error is

$$TE_n^{SDE_4}=\left(\frac{1}{864}y_n^{(7)}-\frac{5}{12}\theta^2\frac{\partial f}{\partial y}(y_n)y_n^{(6)}\right)h^7+\mathcal{O}(h^8),$$

so that the scheme is of order $p=6$. Analogously to Proposition 5.4, we have proved that the above scheme is A_0-stable if $\theta \geq 0$. In addition, the scheme is stiffly stable for some values of θ. This conclusion derives from the analyse of the stability domain boundaries. For example, in Figure 5.5, there are plotted the root locus curves for some specific θ.

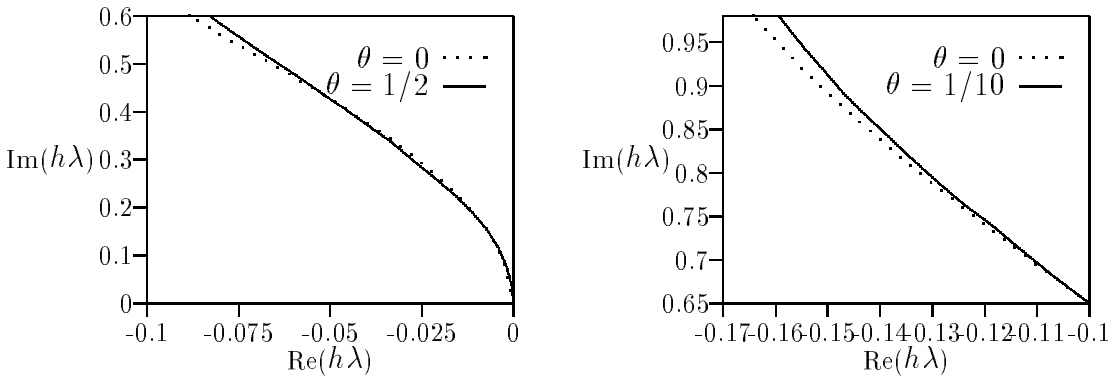

Figure 5.5: Root locus curves for the split ENRIGHT scheme with $k=4$ steps and $b\neq 0$

From these curves we can see that the stability angle is improved when θ increases. Note that the classical method corresponds to $\theta=0$. In Table 5.2 we present the values of the stability angles for some θ values.

Table 5.2: Stiff characteristics for SES with $k=4$ and $b\neq 0$

θ	D	α
0	0.233	80.469
1/20	0.192	80.528
1/10	0.164	80.635
1/2	0.163	81.615

Remarks.

1. If we pose the condition of the dominance of c_{k+3} in the coefficient error, then $\theta \leq 1/19$. For those θ small values, the improvement in the stability angle is not so large. For this reason, in the cases $k \geq 5$ the significant improvement of stability angle is possible only when the error constant is dominated by θ^2.

2. We expect that, at least for $k \leq 7$, the split ENRIGHT scheme with k steps to be stiffly stable for some positive real values θ which depend on k. The proof of the A_0-stability of the general scheme for $\theta \geq 0$ is not easy to be performed and remains an open problem.

5.2.4 Formula with both split derivatives. Split second derivative backward differentiation schemes (SSBDF)

The proposed scheme referred to as the *split second derivative backward differentiation scheme*

$$\boxed{\begin{array}{l}(P)\quad \overline{\alpha}_k y_n + \sum_{i=1}^{k}(\alpha_{k-i} + \overline{\alpha}_{k-i}\theta)y_{n-i} + \\ \qquad +h(\beta_k - d\theta)f(t_n, \overline{y}_n) = h^2(\gamma_k - \theta)g(t_n, \overline{y}_n), \\ (C)\quad \alpha_k y_n + \sum_{i=1}^{k}\alpha_{k-i} f(t_{n-i}, y_{n-i}) + \\ \quad +h(\beta_k - b)f(t_n, y_n) + b\theta f(t_n, \overline{y}_n) = h^2(\gamma_k - \theta)g(t_n, y_n) + h^2\theta g(t_n, \overline{y}_n),\end{array}} \tag{5.19}$$

where α_{k-i}, β_k, γ_k are the k-step second derivative backward differentiation formula coefficients, and $\overline{\alpha}_{k-i}$ are obtained from the maximum order conditions of the predictor formula. Note SSBDF_k the above scheme.

Order

The classical second derivative backward differentiation formula with k steps is of $k+1$ order. Note with c_{k+2} the error constant of this formula.

From the accuracy condition of the corrector formula, we get a value for the unknown variable b. Supposing that the predictor formula has the maximum order k, the order algebraic conditions lie in the system

$$V^T\overline{\alpha} = -2e_2,$$

where

$$v_{ij} = \begin{cases} i^{j+1}, & j \geq 2, \\ 1, & j = 1, \end{cases}$$

are the elements of an inversable matrix, $\overline{\alpha} = (\overline{\alpha}_{k-1}, \ldots, \overline{\alpha}_0)^T$, $e_2 = (0, 1, 0, \ldots, 0)^T$ and

$$b = -\sum_{i=1}^{n}\overline{\beta}_k.$$

Since V is an inversable matrix, the system has an unique solution. The vector $\overline{\alpha}$ and the coefficient b are presented in Table 5.3 for each k which satisfies $3 \leq k \leq 6$.

The split scheme has the same order like the classical ENRIGHT's formula.

Table 5.3: Coefficients $\overline{\alpha}_{k-i}$ of the predictor formula of k order

k	b	$\overline{\alpha}_1$	$\overline{\alpha}_2$	$\overline{\alpha}_3$	$\overline{\alpha}_4$	$\overline{\alpha}_5$	$\overline{\alpha}_6$
3	$\frac{12}{11}$	$\frac{19}{11}$	$-\frac{26}{11}$	$\frac{7}{11}$			
4	$\frac{42}{30}$	$\frac{92}{30}$	$-\frac{159}{30}$	$\frac{84}{30}$	$-\frac{17}{30}$		
5	$\frac{225}{137}$	$\frac{3799}{822}$	$-\frac{7909}{822}$	$\frac{1031}{137}$	$-\frac{2491}{822}$	$\frac{415}{822}$	
6	$\frac{2806272}{1524091}$	$\frac{48208192}{7620455}$	$-\frac{23532768}{1524091}$	$\frac{24304896}{1524091}$	$-\frac{14624064}{1524091}$	$\frac{4862592}{1524091}$	$-\frac{3641472}{7620455}$

Proposition 5.5. *The split second derivative backward differentiation scheme with k steps has $k+1$ order.*

Proof. The local truncation error of the predictor formula has the form

$$\overline{TE}_n^{SSBDF} = \frac{\overline{c}_k}{\alpha_k}\theta h^k y^{(k+1)}(t_n) + \mathcal{O}(h^{k+2}),$$

where $\overline{c}_k$ can be expressed like a function on α_{k-i} and b coefficients. The local truncation error of the split scheme is given by the following equation

$$TE_n^{SSBDF} = \frac{h^{k+2}}{\alpha_k}\left(c_{k+2}y^{(k+2)}(t_n) + \theta^2\overline{c}_k\frac{\partial f}{\partial y}(t_n, y(t_n))y^{(k+1)}(t_n)\right) + \mathcal{O}(h^{k+3})$$

Hence, the split scheme order is $p = k+1$.□

Stability

The classic second derivative backward differentiation method with k steps is A-stable for $k = 1, 2, 3$ and stiffly stable for $k \leq 10$. Hence, the improvement of the stability angle is important only in the case $k \geq 4$.

Case $k = 4$. The split second derivative backward differentiation scheme with four steps has the following equations:

$$(P) \qquad -\frac{415}{144}\overline{y}_n + \left(4 + \frac{92}{30}\theta\right)y_{n-1} - \left(\frac{3}{2} + \frac{159}{30}\theta\right)y_{n-2} + \left(\frac{4}{9} + \frac{84}{30}\theta\right)y_{n-3} -$$

$$- \left(\frac{1}{16} + \frac{17}{30}\theta\right)y_{n-4} + \left(\frac{25}{12} - \frac{42}{30}\theta\right)hf(t_n, \overline{y}_n) = h^2\left(\frac{1}{2} - \theta\right)g(t_n, \overline{y}_n),$$

$$(C) \qquad -\frac{415}{144}y_n + 4y_{n-1} - \frac{3}{2}y_{n-2} + \frac{4}{9}y_{n-3} - \frac{1}{16}y_{n-4} + \left(\frac{25}{12} - \frac{42}{30}\theta\right)hf(t_n, y_n) +$$

$$+\frac{42}{30}hf(t_n, \overline{y}_n) = h^2\left(\frac{1}{2} - \theta\right)g(t_n, y_n) + h^2\theta g(t_n, \overline{y}_n), \tag{5.20}$$

The scheme order is five. The predictor order is only four:

$$\overline{TE}_n^{SSBDF_4} = -\frac{144}{415}\theta\frac{83}{150}h^5 y^{(5)}(t_n) + \mathcal{O}(h^6),$$

The local truncation error produced by the scheme has the form

$$TE_n^{SSBDF_4} = \frac{144}{415}\left[\frac{1}{30}y^{(6)}(t_n) - \frac{83}{150}\theta^2\frac{\partial f}{\partial y}(t_n, y(t_n))y^{(5)}(t_n)\right]h^6 + \mathcal{O}(h^7),$$

The split scheme is A_0-stable at least for $\theta \leq 0$. The proof of this statement is similar with that of Proposition 5.5.

We have plotted the root locus curves of the split scheme in Figure 5.6. The curves are symmetrically to the real axis.

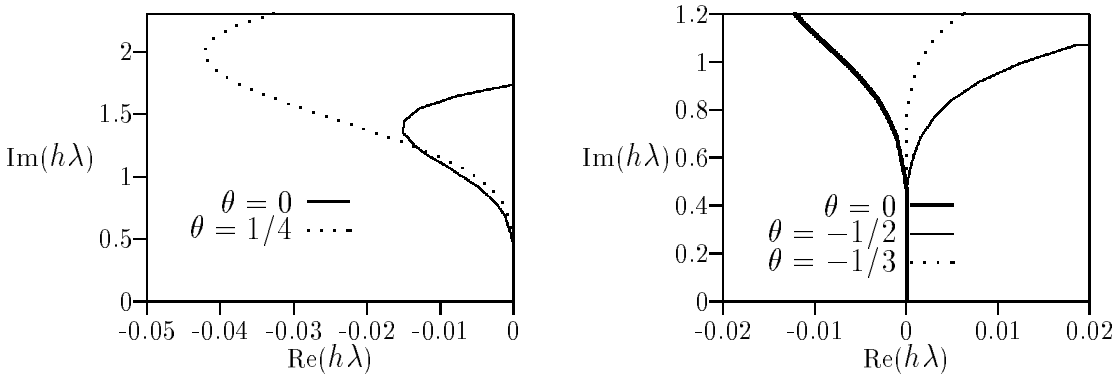

Figure 5.6: Root locus curves for split second derivative backward differentiation scheme with $k = 4$ and $b \neq 0$

The curves show that, in the condition of A_0-stability, the θ corresponding schemes are stiffly stable. Thus, the schemes with $\theta \leq 0$ are stiffly stable. In addition, for some θ values we can reach the A-stability property. We can note the increase of the stability domains with the decrease of the θ values, and conclude that the scheme with $\theta \leq -1/3$ are A-stable (breaking the DAHLQUIST's barrier).

If we put the condition that the error constant of the scheme to be dominated by the error constant of the classical method, then it results $|\theta| \leq 0.245$. For the maximum value, we get stiff-stability with an angle closed to $\pi/2$.

Case $k = 5$. The equations are

$$-\frac{12019}{3600}\overline{y}_n + \left(5 + \frac{3799}{822}\theta\right)y_{n-1} - \left(\frac{5}{2} + \frac{7909}{822}\theta\right)y_{n-2} +$$

$$(P) \qquad + \left(\frac{10}{9} + \frac{1031}{137}\theta\right)y_{n-3} - \left(\frac{5}{16} + \frac{2491}{822}\theta\right)y_{n-4} + \left(\frac{1}{25} + \frac{415}{822}\theta\right)y_{n-5} +$$

$$+\left(\frac{137}{60}-\frac{225}{137}\theta\right)f(t_n,\overline{y}_n)=\left(\frac{h^2}{2}-\theta\right)g(t_n,\overline{y}_n),$$

$$(C)\ -\frac{12019}{3600}y_n+5y_{n-1}-\frac{5}{2}y_{n-2}+\frac{10}{9}y_{n-3}-\frac{5}{16}y_{n-4}+\frac{1}{25}y_{n-5}+\left(\frac{137}{60}-\right.$$

$$\left.-\frac{225}{137}\theta\right)f(t_n,y_n)+\frac{225}{137}\theta f(t_n,\overline{y}_n)=\left(\frac{h^2}{2}-\theta\right)g(t_n,y_n)+h^2\theta g(t_n,\overline{y}_n),$$

The scheme is of six order, with a predictor formula of five order:

$$\overline{TE}_n^{SSBDF_5}=-\frac{3600}{12019}\theta\frac{12019}{24660}h^6y^{(6)}(t_n)+\mathcal{O}(h^7),$$

and the local truncation error

$$TE_n^{SSBDF_5}=\frac{3600}{12019}\left[\frac{1}{42}y^{(7)}(t_n)-\frac{12019}{24660}\theta^2\frac{\partial f}{\partial y}(t_n,y(t_n)y^{(6)}(t_n)\right]h^7+\mathcal{O}(h^8)$$

The scheme is A_0-stable for $\theta\leq 0$ and stiffly stable for some values of the θ parameter. The root locus curves are plotted in Figure 5.7, to prove the statement of the stiff-stability.

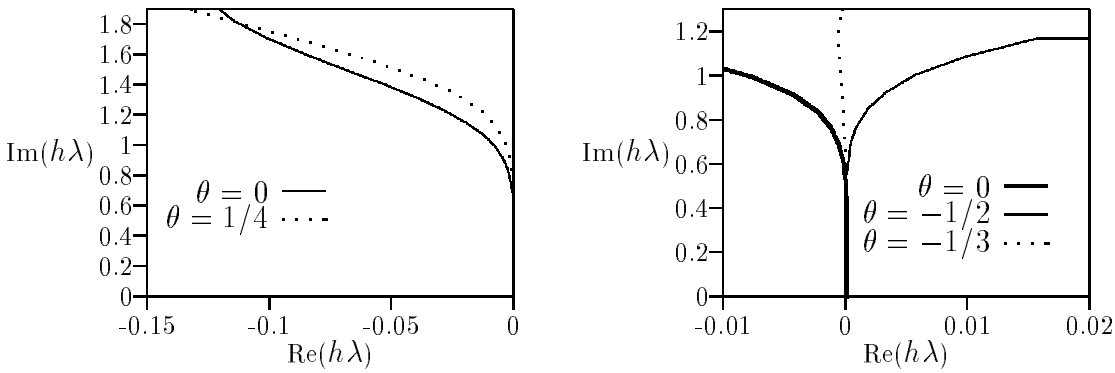

Figure 5.7: Root locus curves for the split second derivative backward differentiation formula with $k=5$ and $b\neq 0$

The curves are symmetrically to the real axis. The stability domains are the regions defined by the root locus curves and the negativ real half-axis. One can see the improvement of the stability angle with the decrease of the θ values. For $\theta=-1/2$, the stability domain of the split scheme corresponds to an A-stable method.

The condition that the error constant be dominated by the error constant of the classical formula is not more restrictive than for the case $k=4$: $|\theta|\leq 0.221$. Therefore, we suppose that the application of the splitting method to the second derivative backward differentiation formula leaves to an improvement of the stability angle not only for $k=4,5$, but also for $6\leq k\leq 10$.

5.2.5 Formula with split second derivative

By splitting only the second derivative from a second derivative backward differentiation formula, we get some stiffly stable methods, but not an improvement of the stability angle.

Example 1. The split ENRIGHT scheme with three steps,

$$\overline{y}_n = y_{n-1} + h\left[\frac{307}{540}f(t_n,\overline{y}_n) + \left(\frac{19}{40}+\frac{5}{2}\theta\right)f(t_{n-1},y_{n-1}) - \left(\frac{1}{20}+\right.\right.$$

$$\left.\left.+4\theta\right)f(t_{n-2},y_{n-2}) + \left(\frac{7}{1080}+\frac{3}{2}\theta\right)f(t_{n-3},y_{n-3})\right] - h^2\left(\frac{19}{180}+\theta\right)g(t_n,\overline{y}_n)$$

$$y_n = y_{n-1} + h\left[\frac{307}{540}f(t_n,y_n) + \frac{19}{40}f(t_{n-1},y_{n-1}) -\right.$$

$$\left.-\frac{1}{20}f(t_{n-2},y_{n-2}) + \frac{7}{1080}f(t_{n-3},y_{n-3})\right] - h^2\left(\frac{19}{180}+\theta\right)g(t_n,y_n) + \theta h^2 g(t_n,\overline{y}_n). \quad (5.21)$$

of five order, a predictor formula of third order,

$$\overline{TE}_n = \theta h^4 y^{(4)}(t_n) + \mathcal{O}(h^5),$$

and the local error

$$TE_n = -\left[\frac{3}{800}y^{(6)}(t_n) + \theta^2\frac{\partial^2 f}{\partial y^2}(t_n,y(t_n))y^{(4)}(t_n)\right]h^6 + \mathcal{O}(h^7),$$

is A_0-stable at least for $\theta \in [-19/360, 19/360]$. Plotting the root locus curves we get the information of the stiff stability for some θ values (see Figure 5.8).

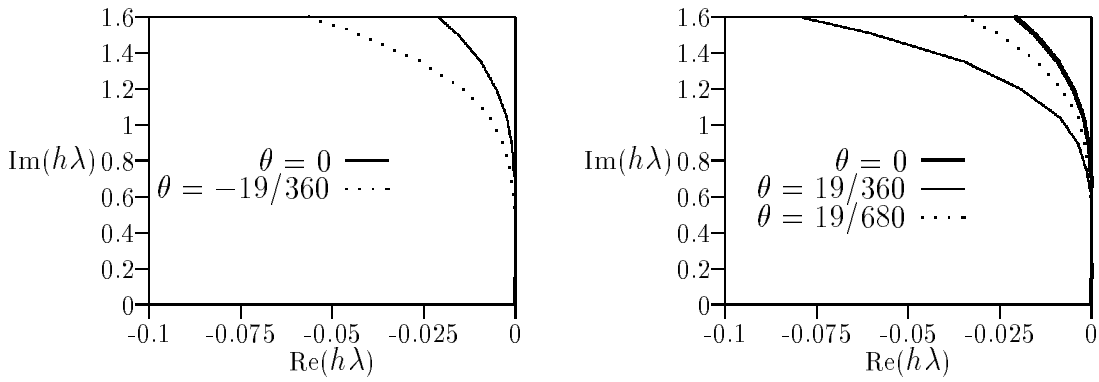

Figure 5.8: Root locus curves for split ENRIGHT scheme with $k = 3$ and $b = 0$

The curves are symmetrically to the real axis. For each specific θ, the stability domains include the regions defined by the plotted locus curves and the real negative

half-axis (from the A_0-stability). Althought the proposed schemes are stiffly stable, the bigest stability domain corresponds to the case of the classical method, $\theta = 0$.

Example 2. The split second derivative backward differentiation scheme with three steps,

$$(P) \qquad -\frac{85}{36}\overline{y}_n + (3-\theta)_{n-1} - \left(\frac{3}{4} - 2\theta\right) + \left(\frac{1}{9} - \theta\right) y_{n-3} + \frac{11}{6} h f(t_n, \overline{y}_n) = \\ = h^2 \left(\frac{1}{2} - \theta\right) g(t_n, \overline{y}_n),$$

$$(C) \qquad -\frac{85}{36} y_n + 3y_{n-1} - \frac{3}{4} y_{n-2} + \frac{1}{9} y_{n-3} + \frac{11}{6} h f(t_n, y_n) = \\ = h^2 \left(\frac{1}{2} - \theta\right) g(t_n, y_n) + h^2 \theta g(t_n, \overline{y}_n),$$

of four order, with a predictor formula of second order

$$\overline{TE}_n = -\frac{36}{85} \theta 2 h^3 y^{(3)}(t_n) + \mathcal{O}(h^4)$$

and the local error

$$TE_n = \frac{36}{84} \left[\frac{1}{20} y^{(5)}(t_n) - 2\theta \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) y^{(3)}(t_n)\right] h^5 + \mathcal{O}(h^6),$$

is A_0-stable at least for $\theta \in [-1, 1/2]$.

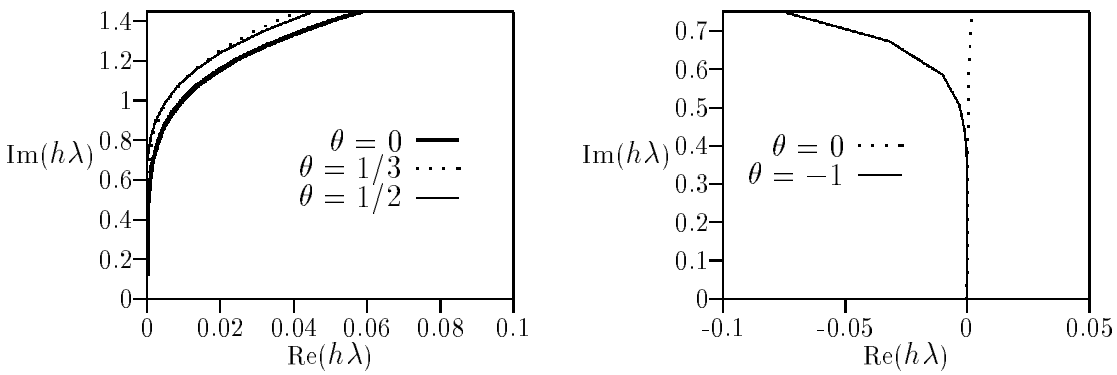

Figure 5.9: Root locus curves for the split second derivative backward differentiation scheme with $k = 3$ and $b = 0$

The plots of the root locus curves (Figure 5.9) for some θ values show that the scheme is stiffly stable for θ in the above mentioned interval. The classical formula corresponds to the case $\theta = 0$.

Chapter 6

EXPONENTIAL FITTED AND HYBRID METHODS

6.1 Parameter dependent formulae

6.1.1 Motivation

When we are applying a linear multistep method to a linear system of differential equations, $y' = Ay$, the error $e_n = y(t_n) - y_n$ depends on the stability of the numerical amplification operator y_n/y_{n-1} and the closeness of this operator to the one of the exact solution. The numerical operator approximates the exact operator in a neighborhood of the origin. Note $\sigma(A)$ the spectrum of the matrix A. Then $h\sigma(A)$ is shrinking into a neighborhood of the origin by making h small enough.

The idea of the exponential fitting method is to replaces the single point of the origin by a set of points in the complex plane, called the fitting points. The exponential fitting method becomes interesting for stiff systems when we observe that fitting points may be very large in magnitude so that h is not required to scale the entire spectrum of A into a neighborhood of the origin.

We use the LINIGER-EHLE's technique, i.e. derive integration formulae with free parameters, others than the stepsize, and then choose these parameters so that a given exponential function satisfies exactly the integration formula.

The fitted formula are:

- backward differentiation formulae;
- second derivative backward differentiation formulae;
- ENRIGHT's second derivative formulae;
- LAMBERT's nonlinear methods.

6.1.2 Exponentially fitted backward differentiation formulae

Formula

We consider the following formulae depending on a real parameter a:

$$\boxed{\sum_{j=0}^{k}(\alpha_j + a\overline{\alpha}_j)y_{n+j-k} = h\beta_k f_n.} \tag{6.1}$$

Order

We get the maxim order $p = k$ when $a = 0$ and the corresponding values α_i are the coefficients of the GEAR's formulae. In the case

$$\boxed{\sum_{j=0}^{k}(\alpha_j + (-1)^i C_k^i a)y_{n+j-k} = h\beta_k f_n.} \tag{6.2}$$

where α_i, β_k are coefficients of the GEAR's formula and $a \neq 0$, the order of accuracy is $k-1$. Note BDF_k^a, the k step a dependent formula and BDF_k, the GEAR's formula.

Examples. We study the case $k \geq 2$, since the case $k = 1$ was studied by MAKULA et al. [83]:

$$\begin{aligned}
k = 2 \quad & (3+a)y_n - 2(2+a)y_{n-1} + (1+a)y_{n-2} = 2hf_n, \\
k = 3 \quad & (11+a)y_n - 3(6+a)y_{n-1} + 3(3+a)y_{n-2} - (2+a)y_{n-3} = 6hf_n, \\
k = 4 \quad & (25+a)y_n - 4(12+a)y_{n-1} + 6(6+a)y_{n-2} - 4(4+a)y_{n-3} + (3+a)y_{n-4} = 12hf_n, \\
k = 5 \quad & (137+a)y_n - 5(60+a)y_{n-1} + 10(30+a)y_{n-2} - 10(20+a)y_{n-3} + \\
& \quad +5(15+a)y_{n-4} - (12+a)y_{n-5} = 60hf_n, \\
k = 6 \quad & (147+a)y_n - 6(60+a)y_{n-1} + 15(30+a)y_{n-2} - 20(20+a)y_{n-3} + \\
& \quad +15(15+a)y_{n-4} - 6(12+a)y_{n-5} + (10+a)y_{n-6} = 60hf_n.
\end{aligned} \tag{6.3}$$

Remark. For $a = 0$ we can get GEAR's formula of order k, i.e.

$$\mathrm{BDF}_k^0 \equiv \mathrm{BDF}_k.$$

If a is chosen so that the last coefficient, depending on this value, is zero, we get the GEAR's $k-1$ step formula. Therefore

$$\mathrm{BDF}_2^{-1} \equiv \mathrm{BDF}_1, \quad \mathrm{BDF}_3^{-2} \equiv \mathrm{BDF}_2, \quad \mathrm{BDF}_4^{-3} \equiv \mathrm{BDF}_3,$$
$$\mathrm{BDF}_5^{-12} \equiv \mathrm{BDF}_4, \quad \mathrm{BDF}_6^{-10} \equiv \mathrm{BDF}_5.$$

Thus, the proposed formulae are some generalization of the well-known GEAR's methods.

Stability

We know that the BDF are stiffly stable. We expect that the above formulae to have this property for other values of a as well.

Proposition 6.1. *[100] For $2 \leq k \leq 6$, the k-step backward differentiation formula of order $k-1$, depending on a real parameter a is stiffly stable for any $a \in (c_k, d_k)$, where $c_k < d_k$ are two specific values. Moreover, for $k = 2$ and $k = 3$, there are two values c'_k and d'_k, where $c'_k < d'_k$, so that the formula is A-stable if $a \in (c'_k, d'_k)$. The constants are*

$$c_k = \begin{cases} -2, k = 2, \\ -5, k = 3, \\ -8, k = 4, \\ -24, k = 5, \\ -15, k = 6, \end{cases} \quad d_k = \begin{cases} \infty, k = 2, \\ 13, k = 3, \\ 10, k = 4, \\ 38, k = 5, \\ 1, k = 6, \end{cases} \quad c'_k = \begin{cases} -2, k = 2, \\ -5, k = 3, \end{cases} \quad d'_k = \begin{cases} 0, k = 2, \\ -1, k = 3. \end{cases}$$

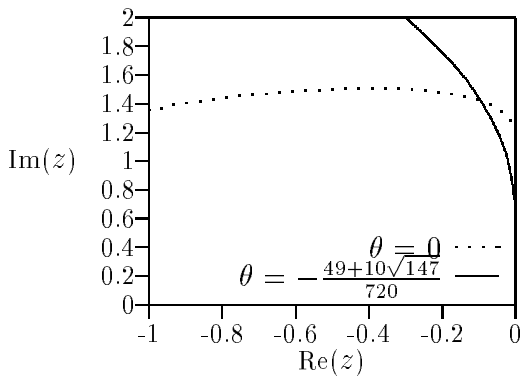

Figure 6.1: Boundaries of the stability domains for BDF_2^a and BDF_3^a

Starting from the characteristic equation, we can get the curves which define the boundary of the stability domains. The curves are plotted for each k in Figure 6.1, 6.2 and 6.3, and are symmetrically to the real axis. The stability domains lie in the exterior of the regions determinated by the plotted curves.

Table 6.1 presents the stiff parameters for the above proposed formulae.

Choosing the a value

Studying the Table 6.0 we can remark that the optimal stability domain is obtained at each k for the smallest a negative value. On an other hand, the error constant of the k step $k-1$ order formula depends on the a value. For some small absolute values, the error constant is smallest than the corresponding constant of the classical formula with $k-1$ steps and $k-1$ order.

Choosing the optimal value for the free parameter, in both sense of stability and accuracy, it is necessary a compromise between the two tendencies: an $|a|$ large value

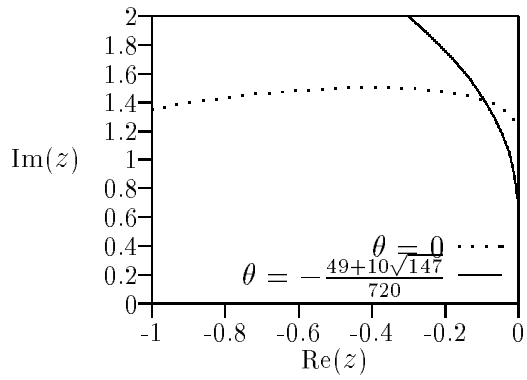

Figure 6.2: Boundaries of the stability domains for BDF_4^a and BDF_5^a

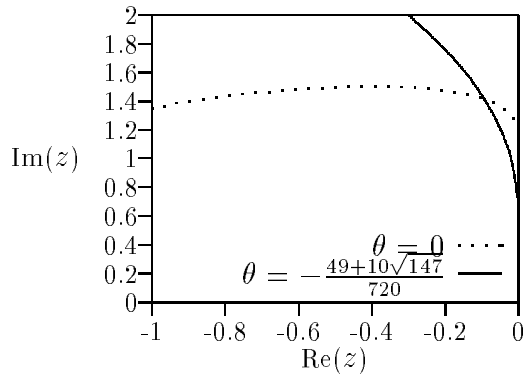

Figure 6.3: Boundaries of the stability domains for BDF_6^a

from the stability condition and a small $|a|$ value from the order condition. Our solutions are mentionated in Table 6.2.

The free parameter a can be used for exponential fitting at some exponential function. The fitted methods are also stiffly-stable.

Proposition 6.2. *[108] For any $2 \leq k \leq 5$, the backward differentiation formula with k steps and $k-1$ order, exponentially fitted to an arbitrary negative real value, is stiffly stable. Moreover, for $k = 2$ and $k = 3$ the formulae are A-stable.*

Table 6.1: Stiff stability parameters for BDF_k^a

$k=2$			$k=3$			$k=4$			$k=5$			$k=6$		
a	D	α	a	D	α	a	D	α	a	D	α	a	D	α
1	0.12	81.7	1	0.29	79.5	1	1	68.5	6	3.44	38.2	1	6.09	15.3
0	0	90	0	0.08	86.0	0	0.67	74.2	0	2.33	51.8	0	5.58	20.0
-0.5	0	90	-1	0	90	-1.5	0.23	82.0	-6	1.22	64.8	-5	3.54	39.3
-1	0	90	-2	0	90	-3	0.08	86.0	-12	0.67	74.2	-10	2.33	51.8
-1.5	0	90	-4	0	90	-6	0.04	86.8	-20	0.42	76.7	-15	1.47	59.9

Table 6.2: The optimal a values for BDF_k^a

k	Order $p=k-1$	Error(BDF_k^a) <Error(BDF_{k-1})	Stiff stability (c_k, d_k)	A-stability (c_k', d_k')
2	1	(-1,1)	[-2,∞)	[-2,0]
3	2	(-2,2)	[-5,13)	[-5,-1]
4	3	(-3,3)	[-8,10)	-
5	4	(-12,12)	[-24,38)	-
6	5	(-10,10)	[-15,1]	-

If the test equation is $y' = \lambda y,\ \lambda \in \mathbf{R}^-$, and the used stepsize is h, then

$$a(q) = \begin{cases} \dfrac{(2q-3)e^{2q} + 4e^q - 1}{(e^q-1)^2}, & k=2, \\ \dfrac{(6q-11)e^{3q} + 18e^{2q} - 9e^q + 2}{(e^q-1)^3}, & k=3, \\ \dfrac{(12q-25)e^{4q} + 48e^{3q} - 36e^{2q} + 16e^q - 3}{(e^q-1)^4}, & k=4, \\ \dfrac{(60q-137)e^{5q} + 300e^{4q} - 300e^{3q} + 200e^{2q} - 75e^q + 12}{(e^q-1)^5}, & k=5. \end{cases} \tag{6.4}$$

where $q = h\lambda$.

Remarks:

1. The k-step exponentially fitted formula at $q=0$ is GEAR's formula of order k.
2. The exponential fitting to $q=-\infty$ reduces the k-step formula to one with $k-1$ steps.
3. We assume that the six-step exponentially fitted formula is also stiffly stable. The numerical calculations are very difficult and tedious, so we omitted this proof.

6.1.3 Exponentially fitted second derivative backward differentiation formulae

Formula

We propose the following iterative process as an a dependent second derivative backward differentiation formula:

$$\boxed{\sum_{j=0}^{k}(\alpha_j + a\overline{\alpha}_j)y_{n+j-k} + h(\beta_k + a\overline{\beta}_k)f_k = h^2\gamma_k g_n.} \tag{6.5}$$

where $g = df/dt$.

Order

We get the maximum order $p = k + 1$ for $a = 0$ and $\alpha_i,\ \beta_k$, the coefficients of the second derivative backward differentiation formula with k steps. In the case

$$\boxed{\sum_{j=0}^{k}(\alpha_j + (-1)^i(ia-1)[1+(k-i)i]/i^2)y_{n+j-k} + h(\beta_k - a)f_k = h\gamma_k g_n.} \tag{6.6}$$

where $\alpha_i, \beta_k, \gamma_k$ are the coefficients of the classical formula, the order of accuracy is k. Note SBDF_k^a, the a-dependent k-step second derivative backward differentiation formula and SBDF_k, the classical second derivative backward differentiation formula.

Examples. Let the following a dependent formulae:

$$k = 2 \qquad \frac{6a-7}{4}y_n + 2(1-a)y_{n-1} + \frac{2a-1}{4}y_{n-2} + \frac{3-2a}{2}hf_n = \frac{1}{2}h^2 g_n,$$

$$k = 3 \quad \frac{66a-85}{36}y_n + 3(1-a)y_{n-1} + \frac{3(2a-1)}{4}y_{n-2} + \frac{1-3a}{9}y_{n-3} + \frac{11-6a}{6}hf_n = \frac{1}{2}h^2 g_n$$

$$k = 4 \qquad \frac{300a-415}{144}y_n + 4(1-a)y_{n-1} + \frac{3(2a-1)}{2}y_{n-2} + \frac{4(1-3a)}{9}y_{n-3} +$$
$$+\frac{4a-1}{16}y_{n-4} + \frac{25-12a}{12}hf_n = \frac{1}{2}h^2 g_n. \tag{6.7}$$

Remark. For each k and $a = 0$ we can get the classical formula with k steps:

$$\mathrm{SBDF}_k^0 \equiv \mathrm{SBDF}_k.$$

If a is chosen so that the last coefficient, depending on this value, is zero, we get the classical formula with $k-1$ steps and k order:

$$\mathrm{SBDF}_2^{1/2} \equiv \mathrm{SBDF}_1, \quad \mathrm{SBDF}_3^{1/3} \equiv \mathrm{SBDF}_2, \quad \mathrm{SBDF}_4^{1/4} \equiv \mathrm{SBDF}_3.$$

Stability

The SBDF$_k$ formulae are stiffly stable for $k \leq 10$. We can try to find some conditions on the parameter a which conserve this property.

Proposition 6.3. *[100] For $2 \leq k \leq 4$, the k-step second derivative backward differentiation formula of k order, depending on a real parameter a is stiffly stable for any $a < a_k$, where*

$$a_k = \begin{cases} 1, & k = 2, \\ 14/15, & k = 3, \\ 5/6, & k = 4. \end{cases}$$

Remark. We assume that, for $k \leq 10$, any k-order k-step second derivative backward differentiation formula is stiffly stable for some a values.

The boundaries of the stability domains are plotted in Figures 6.4 and 6.5. The curves are symmetrically to the real axis.

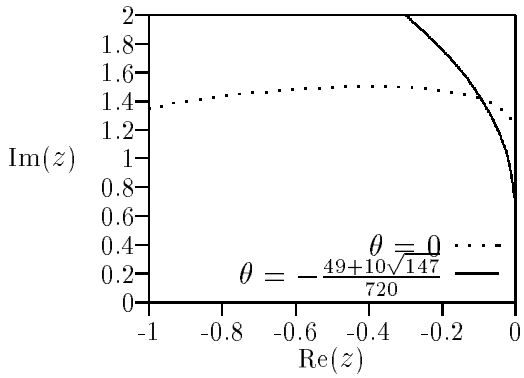

Figure 6.4: Boundaries of the stability domains for SBDF$_2^a$ and SBDF$_3^a$

The stiff stability parameters are mentionated in Table 6.3.

Table 6.3: Stiff stability parameters for SBDF$_k^a$

$k=2$			$k=3$			$k=4$		
a	D	α	a	D	α	a	D	α
-1	0.252	80.794	-1	0.464	77.741	-1	0.845	72.500
0	0	90	0	0	90	0	0.015	89.376
1/4	0	90	1/6	0	90	1/8	0	90
1/2	0	90	1/3	0	90	1/4	0	90
3/4	0	90	2/3	0	90	1/2	0	90

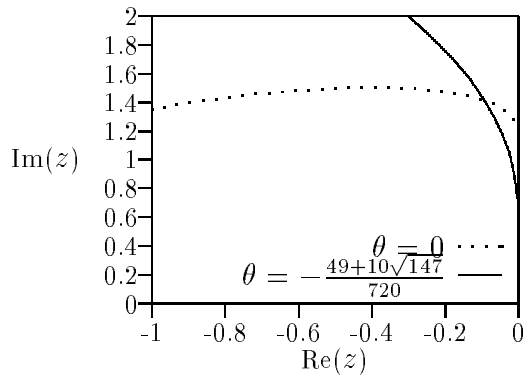

Figure 6.5: Boundaries of the stability domains for SBDF_4^a

Choosing the a values

Similar to the case of the backward differentiation formulae, we can get an improvement of the stability parameters when a increases and the local error descreases when $|a|$ descreases. Thus, it is justified to use the k step formula in the place of the classical formula with $k-1$ steps, if a is chosen like as in Table 6.4.

Table 6.4: The optimal a values for SBDF_k^a

k	Order $p=k$	Error(SBDF_k^a) <Error(SBDF_{k-1})	Stiff-stability $a \in (-\infty, a_k)$	A-stability
2	2	(-1/2,1/2)	($-\infty$,1)	[0,1)
3	3	(-1/3,1/3)	($-\infty$,14/15)	[0,14/15)
4	4	(-25/12,25/12)	($-\infty$,5/6)	[1/8,5/6)

The existence of a free parameter in the condition of stiff stability make possible to apply the exponential fitting method. We expect that the property of stiff stability is extended to the fitted formulae.

Proposition 6.4. *[108] If $2 \leq k \leq 4$, then the second derivative backward differentiation formula with k steps and k order, exponentially fitted to an arbitrary real negative value, is stiffly stable.*

If the test equation is $y' = \lambda y,\ \lambda \in \mathbf{R}^-$, and the used stepsize is h, then

$$a(q) = \begin{cases} \frac{1}{2}\frac{(2q^2-3q+7)e^{2q}-8e^q+1}{(3-2q)e^{2q}-4e^q+1}, & k=2, \\ \frac{1}{6}\frac{(18q^2-66q+85)e^{3q}-108e^{2q}+72e^q-4}{(11-6q)e^{3q}-18e^{2q}+9e^q-2}, & k=3, \\ \frac{1}{12}\frac{(72q^2-300q+415)e^{4q}-576e^{3q}+216e^{2q}-64e^q+9}{(25-12q)e^{4q}-48e^{3q}+36e^{2q}-12e^q+3}, & k=4. \end{cases} \tag{6.8}$$

where $q = h\lambda$.

Remarks:

1. The k-step exponentially fitted formula at $q = 0$ is the second derivative backward differentiation formula of order $p = k + 1$.
2. The exponential fitting to $q = -\infty$ reduces the k-step formula to one with $k-1$ steps and with the same order.
3. We assume that, for any $3 \le k \le 10$, the exponentially fitted formulae are also stiffly stable.

6.1.4 Exponentially fitted ENRIGHT second derivative formulae

Formula

We propose the a dependent ENRIGHT second derivative formula:

$$\boxed{y_n = y_{n-1} + h\sum_{j=0}^{k}(\beta_j + a\overline{\beta}_j)y_{n+j-k} + h^2\gamma_k g_n.} \tag{6.9}$$

where $g = df/dt$.

Order

We get the maximum order $p = k + 2$ for $a = 0$ and $\beta_i,\ \gamma_k$, the coefficient of the ENRIGHT's formula. In the case

$$\boxed{y_n = y_{n-1} + h\sum_{j=0}^{k}(\beta_j - (-1)^i C_k^i \frac{1}{i}a)y_{n+j-k} + h^2\gamma_k g_n.} \tag{6.10}$$

where β_i, γ_k are the coeficcients of the ENRIGHT's formula, the order is $k + 1$. Note ESDF_k^a the k step a dependent formula, and ESDF_k, the ENRIGHT's formula with k steps.

Examples:

$$k=1 \qquad y_n = y_{n-1} + h\left[\left(\frac{2}{3}-a\right)f_n + \left(\frac{1}{3}+a\right)f_{n-1}\right] - \left(\frac{1}{6}-a\right)h^2 g_n,$$

$$k=2 \qquad y_n = y_{n-1} + h\left[\left(\frac{29}{48}-\frac{3}{2}a\right)f_n + \left(\frac{5}{12}+2a\right)f_{n-1} - \left(\frac{1}{48}+\frac{1}{2}a\right)f_{n-2}\right] - \\ - \left(\frac{1}{8}-a\right)h^2 g_n,$$

$$k=3 \qquad y_n = y_{n-1} + h\left[\left(\frac{307}{540}-\frac{11}{6}a\right)f_n + \left(\frac{19}{40}+3a\right)f_{n-1} - \left(\frac{1}{20}+\frac{3}{2}a\right)f_{n-2} + \right. \\ \left. + \left(\frac{7}{1080}+\frac{1}{3}a\right)f_{n-3}\right] - \left(\frac{19}{180}-a\right)h^2 g_n. \tag{6.11}$$

Remark. For $a = 0$ we can get the ENRIGHT's formula of order $k+2$:

$$\mathrm{ESDF}_k^0 \equiv \mathrm{ESDF}_k.$$

If a is chosen so that the last coefficient depending on this value is zero, we get the ENRIGHT's formula with $k-1$ steps and order $k-1$:

$$\mathrm{ESDF}_2^{-1/24} \equiv \mathrm{ESDF}_1, \qquad \mathrm{ESDF}_3^{-7/360} \equiv \mathrm{ESDF}_2.$$

Stability

The ENRIGHT's formulae are stiffly stable for $k \leq 7$. The question is whether the formula with variable coefficients is also stiffly stable.

Proposition 6.5. *[100] For $1 \leq k \leq 3$, the $k+1$ order k-step* ENRIGHT *second derivative formula, depending on a real parameter a, is stiffly stable for any $a < a_k$, where*

$$a_k = \begin{cases} 1/6 \ , & k=1, \\ 1/24 \ , & k=2, \\ 1/144, & k=3. \end{cases}$$

The boundaries of the stability domains are plotted in Figures 6.6 and 6.7. The stability domains include the unbounded sets of complex points from the left half-plane limited by the plotted curves.

The stiff stability parameters are dependent on the a parameter (see Table 6.5).

Remark. We assume that the formula with k-steps and a free parameter a is stiffly stable also if $4 \leq k \leq 7$.

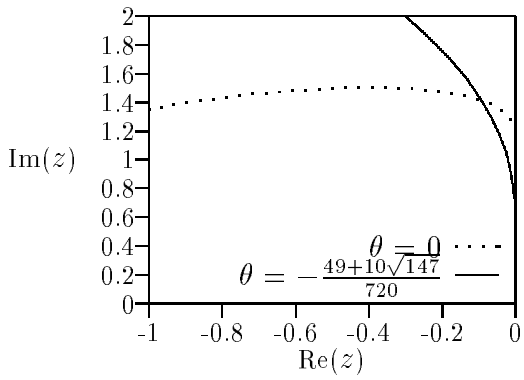

Figure 6.6: Boundaries of the stability domains for ESDF_1^a and ESDF_2^a

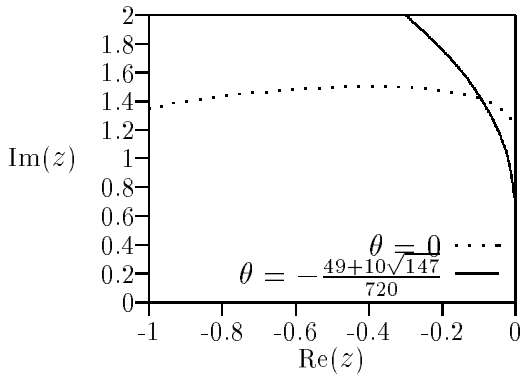

Figure 6.7: Boundaries of the stability domains for ESDF_3^a

Choosing an a value

The stiff stability parameters are optimal for the smallest negative a values and the local error decreases with the decrease of the $|a|$ value. A compromise can be proposed (see Table 6.6).

The free parameter a can be use by the exponential fitting method. We expect that the exponentially fitted formulae are also stiffly stable.

Proposition 6.6. *[108] For $1 \leq k \leq 3$, the* ENRIGHT *second derivative formula of $k+1$ order and with k steps, exponentially fitted to an arbitrary negative real value, is stiffly stable.*

Table 6.5: Stiff stability parameters for ESDF_k^a

$k=1$			$k=2$			$k=3$		
a	D	α	a	D	α	a	D	α
1/12	0.086	86.241	1/48	0.077	87.850	1/288	0.162	87.010
0	0	90	0	0	90	0	0.098	87.922
-1/6	0	90	-1/48	0	90	-7/720	0	90
-1/3	0	90	-1/24	0	90	-7/360	0	90
-1	0	90	-1	0	90	-1	0	90

Table 6.6: Optimal a values for ESDF_k^a

k	p	Error(ESDF_k^a) <Error(ESDF_{k-1})	Stiff stability $a \in (-\infty, a_k)$	A-stability
1	2	(-1/3,1/3)	(-∞,1/6)	(-∞,0]
2	3	(-1/24,1/24)	(-∞,1/24)	(-∞,0]
3	4	(-7/360,7/360)	(-∞,1/144)	(-∞,-7/720]

If the test equation is $y' = \lambda y,\ \lambda \in \mathbf{R}^-$, and the used stepsize is h, then

$$a(q) = \begin{cases} \dfrac{(q^2-4q+6)e^q - 2q - 6}{6[q(q-1)e^q + q]}, & k=1 \\ \dfrac{(6q^2-29q+48)e^{2q} - (20q+48)e^q + q}{24[q(2q-3)e^{2q} + 4qe^q - q]}, & k=2 \\ \dfrac{(114q^2-614q+1080)e^{3q} - (513q+1080)e^{2q} - 54qe^q - 7q}{180[q(6q-11)e^{3q} + 18qe^{2q} - 9qe^q + 2q]}, & k=3 \end{cases} \quad (6.12)$$

where $q = h\lambda$.

Remarks:

1. The k-step exponentially fitted formula at $q = 0$ is the ENRIGHT's formula of order $k+2$.

2. The exponential fitting to $q = -\infty$ reduces the k-step formula to one with $k-1$ steps.

6.1.5 Exponentially fitted multistep nonlinear formulae

Suppose that the initial value problem function has an autonomous form.

The LAMBERT's nonlinear formulae are built up using some rational interpolation functions, instead of the classical polynomial interpolation function.

The error problems encoutered by the nonlinear methods can be diminuated using the exponential fitting method. We mention some results in this direction.

Example 1. Let the rational function

$$I(t) = \frac{At + B}{t + C},$$

and the conditions for an iterative process which modelates the solution of the initial value problem

$$y_n = I(t_n), \quad y_{n-1} = I(t_{n-1}), \quad f_{n-1} = I'(t_{n-1})$$

If we eliminate the coefficients t, B and C, we get an equation in the unknows y_{n-1}, y_n, f_{n-1} and A. Supposing that the numerical solution is exactly integrating the equation $y' = \lambda y$, i.e. $y_n = e^{h\lambda} y_{n-1}$, we get the value A and the iterative process

$$\boxed{y_n = y_{n-1} + \frac{h y_{n-1} f_{n-1}}{y_{n-1} - h\dfrac{q + 1 - e^q}{q(1 - e^q)} f_{n-1}}, \quad q = h\lambda.} \tag{6.13}$$

The method is an onestep one and is of order

$$p_1 \begin{cases} \geq 1, y_{n-1} \neq 0 \\ = 0, \text{otherwise.} \end{cases}$$

For $q \to -\infty$ the f_{n-1} coefficient is one and we get the first LAMBERT's formula, $(L1)$.

A most important property is decribed in the following proposition.

Proposition 6.7. *[103] The formula (6.13) is strongly A-stable for any* $q \in \mathbf{R}^*_-$.

Note that the formula is an explicit one and it is A-stable, i.e. it breaks the DAHLQUIST's second barrier.

Example 2. Let the rational function

$$I(t) = \frac{At + B}{t^2 + Ct + D}$$

and the conditions

$$y_{n-i} = I(t_{n-i}), \quad i = 0, 1, 2, \qquad f_{n-1} = I'(t_{n-1}).$$

Eliminating t, B, C, D we get an equation on y_{n-2}, y_{n-1}, y_n, f_{n-1} and A. Let $A = ch y_{n-1}$. Supposing that the numerical solution is exactly for a scalar test equation, we get the iterative process

$$\boxed{y_n = y_{n-1} + y_{n-1} \frac{(y_{n-1} - y_{n-2})(ch f_{n-1} - y_{n-1}) + 2h y_{n-2} f_{n-1}}{y_{n-1}^2 + 2c y_{n-1}(y_{n-1} - y_{n-2}) - h f_{n-1}(c y_{n-1} + 2y_{n-2})},} \tag{6.14}$$

where

$$c \equiv c(q) := \frac{2q - (e^q - e^{-q})}{2(e^q + e^{-q}) - q(e^q - e^{-q}) - 4}.$$

The method is a two step one and of the order

$$p_2 \begin{cases} = 0, y_{n-1} = f_{n-1} = 0 \\ \geq 2, \text{altfel.} \end{cases}$$

For extremely large q values $c \approx 0$ since $\lim_{q \to -\infty} c(q) = 0$.

Proposition 6.8. *[96] The formula (6.14) is strongly A-stable for any* $q \in \mathbf{R}_-^*$.

Example 3. Let the conditions

$$I(t) = \frac{At + B}{t^2 + Ct + D},$$

$$y_{n-i} = I(t_{n-i}), \quad i = 0, 1, \qquad f_{n-1} = I'(t_{n-1}), \qquad f'(t_{n-1}) = I"(t_{n-1}).$$

Supplimentary, we suppose that the numerical solution is exact for a test scalar equation. Then

$$\boxed{\begin{array}{l} y_n = y_{n-1} + \\ \quad + h y_{n-1} \dfrac{(2f_{n-1} + hf'_{n-1})y_{n-1} + 2h(c-1)f_{n-1}^2}{2y_{n-1}^2 + h[c(2f_{n-1} - hf'_{n-1}) - (2f_{n-1} + hf'_{n-1})]y_{n-1} + 2h^2 f_{n-1}^2}, \end{array}} \tag{6.15}$$

where

$$c \equiv c(q) := \frac{q(q-2)e^q + 2(e^q - 1)}{q(q-2)e^q + q(q+2)}.$$

This is an onestep method and has the order of accuracy

$$p_3 \begin{cases} = 1, f_{n-1} = 0, \\ \geq 2, \text{altfel.} \end{cases}$$

Two special cases can be distinguished. For the first one, we get by exponential fitting at $-\infty$ (stability at infinity), since $\lim_{q \to -\infty} c(q) = 0$. The second one is a LAMBERT's formula, $(L3)$: we get it by exponential fitting at zero.

The exponential fitting to an arbitrary negative real value q lies also to a strongly A-stable formula.

Example 4. Let the rational function

$$I(t) = \frac{At^2 + Bt + C}{t^2 + Dt + E},$$

the conditions

$$y_{n-i} = I(t_{n-i}), \quad i = 0, 1, 2, \qquad f_{n-j} = I'(t_{n-j}), \quad j = 1, 2$$

and the request of the identity between the numerical solution and the exact solution of the equation $q = h\lambda$. Then

$$y_n = u_n / v_n, \tag{6.16}$$

$$u_n = y_{n-1}(y_{n-1} - y_{n-2})(4cy_{n-1} - (c-1)y_{n-2}) + 2h(1-c)y_{n-1}^2 f_{n-2} -$$

$$-hy_{n-2}(y_{n-2}+cy_{n-1})f_{n-1}-2h^2cy_{n-1}f_{n-1}f_{n-2},$$
$$v_n=(3c+1)y_{n-1}(y_{n-1}-y_{n-2})+2h(1-c)y_{n-1}f_{n-2}-h(y_{n-2}+cy_{n-1})f_{n-1}-$$
$$-2h^2f_{n-1}f_{n-2},$$

where

$$c=\frac{e^3q-(2q^2-q+1)e^{2q}-(2q+1)e^q+q+1}{(q-3)e^{3q}+(2q+7)e^{2q}-(2q^2+3q+5)e^q+1}.$$

The formula has maximum third order. The exponential fitting at $-\infty$ lies to the formula

$$y_n=y_{n-1}+\frac{(y_{n-1}-y_{n-2})^2+h(y_{n-1}-y_{n-2})f_{n-1}-2h^2f_{n-1}f_{n-2}}{2(y_{n-1}-y_{n-2})-h(2f_{n-2}-f_{n-1})}. \tag{6.17}$$

Example 5. A formula with good implementation results is the following:

$$\boxed{y_n=y_{n-1}+\frac{(y_{n-1}-y_{n-2})f_{n-1}^2}{f_{n-1}^2-(y_{n-1}-y_{n-2})f'_{n-1}}.} \tag{6.18}$$

The formula is a two step one and has maximum third order. It is exactly for any test equation $y'=\lambda y$, if the first iteration produced by the starting procedure is exactly: note $w_n=y_n/y_{n-1}$, we get

$$w_n=1+\frac{(1-w_{n-1}^{-1})\lambda^2}{\lambda^2-(1-w_{n-1}^{-1})\lambda^2}=w_{n-1}$$

From this equality we get the A-stability property for the proposed formula.

6.2 Onestep hybrid methods

6.2.1 Motivation

Introducing some extradivision solution evaluations in a linear multistep formula, the hybrid methods allow the breaking of DAHLQUIST's second barrier. The onestep hybrid methods are closed to the form of RUNGE-KUTTA's methods. The actual variants do not request a large calculus volume, but, for high stability properties, are supposing that the extradivision values depend on y_n.

A particular class of such methods is due to ENGLAND[37]. The basic idea is to maintain the same stability properties like for ENRIGHT's second derivative formulae: the second derivative evaluation is replaced by an evaluation of the solution in an extradivision point.

Note that, for the onestep case, the ENGLAND's formula of third order has the following form:

$$\begin{cases} y_{n+1}=y_n+h\dfrac{3\theta-1}{6\theta}f_n+h\dfrac{3\theta-2}{6(\theta-1)}f_{n+1}-\dfrac{h}{6\theta(\theta-1)}f_{n+\theta}, \\ y_{n+\theta}=(\theta-1)^2y_n-\theta(\theta-2)y_{n+1}+h\theta(\theta-1)f_{n+1}. \end{cases}$$

One can get the particular formula studied by ENGLAND from the condition of a zero coefficient for f_n in the first equation:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{4} f_{n+1} + \dfrac{3h}{4} f_{n+1/3}, \\ y_{n+1/3} = \dfrac{5}{9} y_{n+1} + \dfrac{4}{9} y_n - \dfrac{2h}{9} f_{n+1}. \end{cases} \tag{6.19}$$

The local discretization error is:

$$TE_n = -\frac{h^4}{216}\left[y^{(4)}(t_n) - 4\frac{\partial^2 f}{\partial y^2}(t_n, y(t_n))y^{(2)}(t_n)\right] + \mathcal{O}(h^5).$$

The scheme is L-stable.

The hybrid methods are some particular cases of *pseudo* RUNGE-KUTTA *methods* (for details see [96]). For example, the BOKHOVEN's methods [119] are represented by the iterative process:

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i, \quad k_i = (1-\theta_i)y_n + \theta_i y_{n+1} + h\sum_{j=1}^{s} a_{ij}k_j, \quad i = 1, \ldots, s.$$

In this θ-class, we distinguish one A-stable scheme of four order:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(f_{n+1} + f_n) + \dfrac{2h}{3} f_{n+\frac{1}{2}}, \\ y_{n+\frac{1}{2}} = \dfrac{1}{2}(y_{n+1} + y_n) - \dfrac{h}{8}(f_{n+1} - f_n). \end{cases} \tag{6.20}$$

The local discretization error is:

$$TE_n = -\frac{h^5}{2880}\left[y^{(5)}(t_n) - 5\frac{\partial^3 f}{\partial y^3}(t_n, y(t_n))y^{(2)}(t_n)\right] + \mathcal{O}(h^6).$$

Another A-stable scheme, of third order, is the folowing:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(f_{n+\theta_1} + f_{n+\theta_2}), \\ y_{n+\theta_1} = \dfrac{2+\sqrt{3}}{6} y_n + \dfrac{4-\sqrt{3}}{6} y_{n+1} - \dfrac{h}{6} f_{n+1}, \\ y_{n+\theta_2} = \dfrac{4-\sqrt{3}}{6} y_n + \dfrac{2+\sqrt{3}}{6} y_{n+1} + \dfrac{h}{6} f_n. \end{cases} \tag{6.21}$$

where $\theta_1 = (3-\sqrt{3})/6$, $\theta_2 = (3+\sqrt{3})/6$, and the local discretization error

$$TE_n = -\frac{h^4}{24}\frac{\partial^2 f}{\partial y^2}(t_n, y(t_n))y^{(2)}(t_n) + \mathcal{O}(h^5).$$

We mention also the OBRECHKOFF's A-stable formula of four order, with second derivative

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n) - \frac{h^2}{12}(f'_{n+1} - f'_n),$$

and the local discretization error:

$$TE_n = \frac{h^5}{720} y^{(5)}(t_n) + \mathcal{O}(h^6).$$

In the following sections we study three formula classes, generalizing the above mentionated scheme in the idea to get some A-stable or L-stable scheme of superior order, with a similar calculus effort.

6.2.2 The first method class

Formula

A generalizing scheme for the ENGLAND's θ-class and the BOCKHOVEN's method (6.20) is the folowing:

$$\boxed{\begin{cases} y_{n+1} = y_n + h\left(\dfrac{v+u}{2} f_{n+1} + \dfrac{v-u}{2} f_n\right) - hv f_{n+\theta}, \\ y_{n+\theta} = w y_{n+1} + (1-w) y_n + h\dfrac{\theta - w + x}{2} f_{n+1} + h\dfrac{\theta - w - x}{2} f_n, \end{cases}} \tag{6.22}$$

which has the minimum order in each equation.

We discuss also the following variant:

$$\begin{cases} y_{n+1} = y_n + \dfrac{1+u}{2} h f\left(v y_{n+1} + (1-v) y_n\right) + \dfrac{1-u}{2} h f_{n+\theta}, \\ y_{n+\theta} = w y_{n+1} + (1-w) y_n + \dfrac{h}{2}(\theta - w + x) f_{n+1} + \dfrac{h}{2}(\theta - w - x) f_n. \end{cases} \tag{6.23}$$

The first equation is referred to as the *quadrature formula*, and the second, as the *interpolation formula.*

We search the order and the stability properties to find some new methods which have atmost the same performances like the methods mentionated in the last section.

Order

Since, in the given form, the order is $p = 1$, the scheme is consistent.

The maximum order for the scheme (6.22) is $p = 4$. The corresponding scheme is the BOKHOVEN's method of four order which has the same stability function like that of the OBRECHKOFF's formula.

The maximum order for the scheme (6.23) is $p = 3$: we request the third order for the quadrature formula and the minimum second order for the interpolation formula. We get two formula classes with a free parameter in the interpolation formula. The fixed coefficients are, for the first class,

$$u = -\frac{1}{2}, \quad x = -\frac{2}{9}, \quad v = 0, \quad \theta = \frac{2}{3},$$

respectively, for the second class,

$$u = -\frac{1}{2}, \quad x = -\frac{2}{9}, \quad v = 1, \quad \theta = \frac{1}{3},$$

where w is the free parameter. If the interpolation formula has only second order, the local discretization error of this formula has influence on the local discretization error of the scheme. If we request the third order for the interpolation formula, the local discretization error has the same error coefficient like the quadrature formula. From the two formula corresponding to the condition of third order for the interpolation formula, only one is A-stable. For this the θ value is $\theta = \frac{2}{3}$ and the scheme has the following form:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{4} f_n + \dfrac{3h}{4} f_{n+2/3}, \\ y_{n+2/3} = \dfrac{20}{27} y_{n+1} + \dfrac{7}{27} y_n - \dfrac{4h}{27} f_{n+1} + \dfrac{2h}{27} f_n. \end{cases} \tag{6.24}$$

The local discretization error is

$$TE_n = \frac{h^4}{216} y^{(4)}(t_n) + \mathcal{O}(h^4).$$

Note that the error constant, $1/216$, is lowest than the ones of the ENRIGHT's second derivative formula, and of the ENGLAND's scheme.

Stability

Let the case of a third order quadrature formula and a second order interpolation formula. The conditions of strong A-stability restrict the w parameter:

$$w > \begin{cases} 2/3, & v = 0 \\ 1/3, & v = 1 \end{cases}.$$

For example, the scheme (6.24) is strongly A-stable and has a third order interpolation formula.

Exponential fitting and stability at infinity

The free parameter w can be used in the purpose of the exponential fitting. If the fitting point is $q \in R_-$, then

$$w(q) = \begin{cases} \dfrac{4}{9} \dfrac{(q^2-6)e^q + 2q^2 + q + 6}{(q^2-2q)e^q + q^2 + 2q}, & v = 0, \\ \dfrac{1}{9} \dfrac{(q^2+6q-24)e^q + 5q^2 + 18q + 24}{(q^2-2q)e^q + q^2 + 2q}, & v = 1. \end{cases}$$

The schemes are strongly A-stable for any $q \in \mathbf{R}_-$. We get some particular schemes by exponential fitting at zero and infinity. The corresponding coefficients are the followings:

$$\lim_{q \to -\infty} w(q) = \begin{cases} 8/9, & v = 0 \\ 5/9, & v = 1 \end{cases}, \qquad \lim_{q \to 0} w(q) = \begin{cases} 2/3, & v = 0 \\ 1/3, & v = 1 \end{cases}.$$

The exponential fitting at $-\infty$ is equivalent with the request of stability at infinity. We get two formulae. The first one, for $w = 5/9$, is the ENGLAND's onestep method. The second one, for $\theta = 2/3,\ w = 8/9$, is the folowing:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{4} f_n + \dfrac{3h}{4} f_{n+2/3}, \\ y_{n+2/3} = \dfrac{8}{9} y_{n+1} + \dfrac{1}{9} y_n - \dfrac{2h}{9} f_{n+1}. \end{cases} \tag{6.25}$$

The local discretization error is

$$TE_n = \frac{h^4}{216} \left[y^{(4)}(t_n) + 2 \frac{\partial^2 f}{\partial y^2}(t_n, y(t_n)) y^{(2)}(t_n) \right] + \mathcal{O}(h^5).$$

The error constant is comparable with those of the ENRIGHT's second derivative formula and the corresponding ENGLAND's onestep method. The scheme has also the L-stability property.

6.2.3 The second method class

Formula

The BOKHOVEN's scheme uses two additional points. The following method, with two extradivision evaluations and of the minimum order one, is a generalization of the BOCKHOVEN's scheme:

$$\boxed{\begin{cases} y_{n+1} = y_n + h \dfrac{1+u}{2} f_{n+\theta_1} + h \dfrac{1-u}{2} f_{n+\theta_2} \\ y_{n+\theta_1} = w_1 y_{n+1} + (1-w_1) y_n + \dfrac{h}{2}(\theta_1 - w_1 + x_1) f_{n+1} + \dfrac{h}{2}(\theta_1 - w_1 - x_1) f_n \\ y_{n+\theta_2} = w_2 y_{n+1} + (1-w_2) y_n + \dfrac{h}{2}(\theta_2 - w_2 + x_2) f_{n+1} + \dfrac{h}{2}(\theta_2 - w_2 - x_2) f_n \end{cases}} \tag{6.26}$$

Order

The maximum order of a such scheme is four. The corresponding method is the following:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2} f_{n+1/2+\sqrt{3}/6} + \dfrac{h}{2} f_{n+1/2-\sqrt{3}/6} \\ y_{n+1/2\pm\sqrt{3}/6} = \left(\dfrac{1}{2} \pm \dfrac{2\sqrt{3}}{9} \right) y_{n+1} + \left(\dfrac{1}{2} \mp \dfrac{2\sqrt{3}}{9} \right) y_n - \dfrac{h}{6} \left(\dfrac{1}{2} \pm \dfrac{\sqrt{3}}{6} \right) f_{n+1} + \\ \qquad + \dfrac{h}{6} \left(\dfrac{1}{2} \mp \dfrac{\sqrt{3}}{6} \right) f_n. \end{cases} \tag{6.27}$$

The local discretization error is

$$TE_n = \frac{h^5}{4320} \left[y^{(5)}(t_n) + 5 \frac{\partial^3 f}{\partial y^3}(t_n, y(t_n)) y^{(2)}(t_n) \right] + \mathcal{O}(h^6).$$

Comparing the error constant with those of the BOKHOVEN's method, and the OBRECHKOFF's formula, we ascertain a substantial improvement.

If we request only a third order for the scheme, we get three free parameters. Let $u,\ w_1,\ w_2$ the free parameters. Hence,

$$\theta_1 = \frac{1}{2} \pm \frac{\sqrt{3(1-u^2)}}{6(1+u)},\ \theta_2 = \frac{1}{2} \mp \frac{\sqrt{3(1-u^2)}}{6(1-u)},\ x_1 = -\frac{1+2u}{6(1+u)},\ x_2 = \frac{2u-1}{6(1-u)}.$$

Stability

The stability function of the method of minimum third order is

$$R(z) = \frac{1+(1-2t)z+(1/3-t)z^2}{1-2tz-(1/6-t)z^2}, \quad t = \frac{1+u}{4}w_1 + \frac{1-u}{4}w_2$$

The condition of strong A-stability requests that $t > 1/4$. The BOKHOVEN's third order scheme is only A-stable, since $t = 1/4$.

If $t = 1/3$, we get a method subclass which depends on two parameters $(w_1,\ w_2)$. All schemes from this subclass have the same stability function, therefore the same stability properties like the ENRIGHT's onestep second derivative formula.

If $w_1 = w_2 = 2/3$, then u is the only one free parameter which can be chosen so that the local discretization error has a minimum constant error.

The stability function of the four order scheme is identically with that of the OBRECHKOFF's formula, and, therefore, the scheme is A-stable.

In the class of third order schemes we can distinguish some L-stable methods. If we request a third order in both equation of the scheme, then one solution has the property of the stability at infinity:

$$\begin{cases} y_{n+1} = y_n + \dfrac{2+\sqrt{3}}{4}hf_{n+\sqrt{3}/3} + \dfrac{2-\sqrt{3}}{4}hf_{n-\sqrt{3}/3}, \\ y_{n+\sqrt{3}/3} = \dfrac{9-2\sqrt{3}}{9})y_{n+1} + \dfrac{2\sqrt{3}}{9}y_n - \dfrac{3-\sqrt{3}}{9}hf_{n+1} + \dfrac{4\sqrt{3}-6}{9}hf_n, \\ y_{n-\sqrt{3}/3} = \dfrac{9+2\sqrt{3}}{9}y_{n+1} - \dfrac{2\sqrt{3}}{9}y_n - \dfrac{3+\sqrt{3}}{9}hf_{n+1} - \dfrac{4\sqrt{3}+6}{9}hf_n. \end{cases} \tag{6.28}$$

The local discretization error is:

$$TE_n = \frac{h^4}{72}y^{(4)}(t_n) + \mathcal{O}(h^5).$$

Note the identity of the constant error of the above scheme and that of the ENRIGHT's onestep second derivative formula. The derivative evaluation is replaced by two function evaluations. The advantage of this scheme compared with the BOKHOVEN's third order scheme consists in the nonexistance of a perturbation in the local discretization formula due to the interpolation formula.

6.2.4 The third method class

Formula

The BOKHOVEN's third order scheme uses three function evaluations at one step of the NEWTON's procedure for solving the implicit equations in the unknowns $y_{n+\theta_1}$, $y_{n+\theta_2}$, y_{n+1}. We can estimate the number of function evaluations to $3m+1$, where m is the medium number of the NEWTON's iterations to one step of the integration interval. Usually, the value m is small, since the convergence order of the NEWTON's iterations is quadratic.

Let the following scheme which also needs three function evaluations at one NEWTON's iteration:

$$\boxed{\begin{cases} y_{n+1} = y_n + ahf(uy_{n+1} + (1-u)y_n) + bhf(y_{n+\theta}) + \\ \qquad +chf(vy_{n+1} + (1-v)y_n), \\ y_{n+\theta} = wy_{n+1} + (1-w)y_n + hdf(uy_{n+1} + (1-u)y_n) + \\ \qquad +hef(vy_{n+1} + (1-v)y_n). \end{cases}} \tag{6.29}$$

Order

The conditions of third order are the folowings:

$$a+b+c=1,\ \theta = w+d+e,\ \theta^2 = w+2(du+ev),\ au+b\theta+cv = \frac{1}{2},$$

$$au^2+b\theta^2+cv^2 = \frac{1}{3},\ au+b\theta^2+cv = \frac{1}{3}.$$

Therefore, the third order scheme class depends on three free parameters. Let u, v, θ these free parameters.

The local discretization error is the sum of the error produced by the quadrature formula,

$$TE_n^{(1)} = -\frac{h^4}{36}\left[\left(\frac{1}{2}-\theta\right)y^{(4)} + uv\frac{3\theta^2-4\theta+1}{\theta^2-\theta}\frac{\partial^3 f}{dy^3}f^3\right](t_n, y(t_n)) + \mathcal{O}(h^5)$$

and the error produced by the interpolation formula (in the approximation of the exact value $y(t_n+\theta h)$),

$$TE_n^{(2)} = \frac{h^4}{36}\left\{\left[(1-\theta)y^{(3)} - \frac{(u+v-1)\theta - [6(\theta^2-\theta)+1]uv}{\theta^2-\theta}\left(\frac{\partial f}{\partial y}\right)^2 f)\right]\frac{\partial f}{\partial y}\cdot\right.$$

$$\left.\cdot f\right\}(t_n, y(t_n)) + \mathcal{O}(h^5).$$

The ENGLAND's onestep method is a particular case of the above mentionated schemes, which has the third order. The maximum order for a class scheme is four, corresponding to the BOKHOVEN's onestep method.

Stability

Getting some L-stable methods, firstly we request the property of stability at infinity for our scheme. Algebraically, this condition means that

$$d(1-u)+e(1-v)=0$$

Hence, the method coefficients are

$$a=\frac{v-\frac{1}{2}}{v-u}+\frac{v-\theta}{6(\theta^2-\theta)(v-u)},\ c=\frac{u-\frac{1}{2}}{u-v}+\frac{u-\theta}{6(\theta^2-\theta)(u-v)},\ b=-\frac{1}{6(\theta^2-\theta)},$$

$$d=\frac{(\theta^2-\theta)(v-1)}{v-u},\ e=\frac{(\theta^2-\theta)(u-1)}{u-v},\ w=2\theta-\theta^2$$

where θ is the solution of the equation

$$[6uv-3(u+v)+3]\theta^2-[6uv-2(u+v)+2]\theta+uv=0.$$

Note that we get a subclass which depends on two parameters, u and v. All the subclass schemes are L-stable, since the stability function is identically with the stability function of the ENRIGHT's second derivative onestep formula.

Examples.

- For $u=0,\ v=1$ or $u=1,\ v=0$, we get the ENGLAND's θ-class.
- The values $v=1,\ \theta=\frac{1}{3}$ lie to the ENGLAND's onestep particular method.
- The values $v=0,\ \theta=\frac{2}{3}$ correspond to a method subclass which depends on the parameter u:

$$\begin{cases} y_{n+1}=y_n+\dfrac{3h}{4}f_{n+2/3}+\dfrac{h}{4}f_n, \\ y_{n+2/3}=\dfrac{8}{9}y_{n+1}+\dfrac{1}{9}y_n-\dfrac{2h}{9u}f(uy_{n+1}+(1-u)y_n)+\dfrac{2(1-u)h}{9u}f_n. \end{cases} \tag{6.30}$$

  For $u=1$ we get the scheme (6.25).

- If $u+v=1$, then the coefficients expressions can be simplified:

$$\theta=\frac{1}{2}\pm\frac{\sqrt{3}}{6},\quad d=\frac{u}{6(u-v)},\ e=\frac{v}{6(u-v)},\ w=\frac{1}{6}+\theta,$$

$$a=\frac{\theta-\frac{1}{2}}{v-u},\ b=1,\ c=\frac{\theta-\frac{1}{2}}{u-v}.$$

  The local discretization error of a such formula is

$$TE_n=-\frac{h^4}{36}\Bigg[\pm\frac{\sqrt{3}}{6}\left(y^{(4)}(t_n)+6u(1-u)\left(\frac{\partial^3 f}{dy^3}f^3\right)(t_n,y(t_n))\right)-$$

$$-\left(\frac{1}{2}\mp\frac{\sqrt{3}}{6}\right)y^{(3)}(t_n)\left(\frac{\partial f}{dy}f\right)(y_n,y(t_n))\Big]+\mathcal{O}(h^5).$$

For simpliffing this expressions, we can request that $1-6u(1-u)=0$. The resulting scheme has the form

$$\left\{\begin{array}{l} y_{n+1}=y_n+hf\left(y_{n+\frac{1}{2}\pm\frac{\sqrt{3}}{6}}\right)\mp\frac{1}{2}hf\left((\frac{1}{2}+\frac{\sqrt{3}}{6})y_{n+1}+(\frac{1}{2}-\frac{\sqrt{3}}{6})y_n\right)\pm \\ \pm\frac{1}{2}hf\left((\frac{1}{2}-\frac{\sqrt{3}}{6})y_{n+1}+(\frac{1}{2}+\frac{\sqrt{3}}{6})y_n\right), \\ y_{n+\frac{1}{2}\pm\frac{\sqrt{3}}{6}}=\left(\frac{2}{3}\pm\frac{\sqrt{3}}{6}\right)y_{n+1}+\left(\frac{1}{3}\mp\frac{\sqrt{3}}{6}\right)y_n- \\ \quad -\frac{1+\sqrt{3}}{12}hf\left(\left(\frac{1}{2}+\frac{\sqrt{3}}{6}\right)y_{n+1}+\left(\frac{1}{2}-\frac{\sqrt{3}}{6}\right)y_n\right)+ \\ \quad +\frac{\sqrt{3}-1}{12}hf\left(\left(\frac{1}{2}-\frac{\sqrt{3}}{6}\right)y_{n+1}+\left(\frac{1}{2}+\frac{\sqrt{3}}{6}\right)y_n\right). \end{array}\right. \tag{6.31}$$

For some system functions, the constant error of this formula can be lowest than the one of the ENRIGHT's onestep formula, which has the same order and stability properties.

Chapter 7

PARALLEL COMPUTING

7.1 Motivation

The problem associated with the stiff ordinary differential equation systems in parallel processing is due to the fact that the calculus can not be started simultaneously on many processors with an explicit formula. We propose two different algorithms. One is based on the method of extrapolation which has stability properties requested by stiff equations. This algorithm can be applied to any kind of stiff systems. The second one is built-up for some special classes of stiff ODE, $y'(t) = A(t)y(t) + g(t)$. The both algorithms have a high efficiency when the system function has many components. The approximation error is equal to the one produced by the analogous sequential algorithm.

Parallelism in solving ODE can be expreseed via three distinct *avenues* (ISERLES, NORSETT [60]):

1. coding a specific method so that it can be performed simultaneously on several processors;

2. splitting variables in a multivariable ODE system between processors;

3. exploiting parallelism in performing the request computer algebra, solving linear and nonlinear algebraic systems of equations.

7.1.1 First avenue

Many serial methods contain a potential degree of parallelism. Most suitable for the parallel implementation are the multivalue methods. The parallel methods from this avenue preserve the stability and accuracy of the basic sequential algorithm.

The predictor-corrector schemes can be easily implemented in a parallel mode. Many exemples are presented by MIRANKER and LINIGER ([87], [88]). The application of those schemes to stiff system is not succesfully because almost all ones are equivalent with some explicit formulas. The stability characteristics can be improved. In this direction we can mention the papers of KATZ et al. [70], WORLAND [125], GHOSHAL et al. [49]. These improvements make possible to handle slightly stiff equations.

The block methods are easily adapted to a parallel mode with no degradation in the accuracy of the solution. The performance of HUTCHINSON and KHALAF parallel implementation [58] is dependent on the number of scheme nodes. CHU and HAMILTON's algorithm [29] has a numerical efficiency depending on the dimension of the solving system and on the complexity of the system function.

ISERLES and NORSETT [60] investigate the degree of the parallelism of the RUNGE-KUTTA methods. The parallelism depends on the exploitation of the sparsity structure of the RUNGE-KUTTA matrix. KARAKASHIAN and RUST's algorithm [69] is designed to solve a linear system of ODE by a RUNGE-KUTTA process. The efficiency of this algorithm depends on the system dimension. For the proposed test system with variable dimension, the parallel mode becames competitive only for a number of components of hundreds order.

The parallel RUNGE-KUTTA method proposed by EVANS and SANUGI [42] makes use of the special form of the method matrix. In an other paper, EVANS and MEGSON [41] consider a systolic array construction of extrapolation tables. The basis schemes are not suitable for the stiff integration.

GALLIGANI and RUGGIERO [44] propose a parallel method which is suitable to be applied to a large set of linear ODE systems. The method has good stability properties, which makes it usefull for the stiff systems integration.

7.1.2 Second avenue

This way request a reconstruction of the class of numerical methods for ODE. The basic idea is due to NIVERGELT [92]. The integration interval is divided into equal subintervals and each processor is responsable for the integration on only one subinterval. To make efficiently the algorithm, the processors must be started at appropriate time and must work simultaneously. For this purpose it is used a starting value (an approximation of the solution at the begining of the processor subinterval). In the special case of a stiff system, the starting value can not be obtained by an explicit method, only in the case when the subintervals number and the step size are very small. In the case of using an implicit starting formula, we reach the class of block methods. An other problem associated with the stiff case is due to the fact that the approximate solution, produced by the parallel algorithm, can be unstable, also when the basic method for integration on each subinterval has good stability properties. The error of the approximate parallel solution is different from that of sequential solution.

7.1.3 Third avenue

KNIRCH [71] studies a scheme for solving the stages algebraic equations of a RUNGE-KUTTA method. The reason for unsuccesfully using of this scheme in the stiff case is the same like for NIVELGELT method.

7.1.4 Proposed algorithms

In the frame of the first avenue, one algorithm explotates the parallelism in the extrapolation methods. It can be applied to any stiff system for which the basic extrapolation method works. The efficiency depends on the dimension of the system, while the degree of the parallelism is function of the requested accuracy of the numerical solution.

The second algorithm is an hybrid one. It can be applied to linear stiff system with variable coefficients. The proposed parallel algorithm divides the responsabilities of the processors on the interval of integration (second avenue), produces the same error like the sequential algorithm (characteristic for the first avenue) and makes use of the linearity of the system (third avenue). The efficiency depends on the complexity of the system function, ans the degree of parallelism, on the dimension of the system.

In both cases, the numerical results show that a high efficiency can be obtained for a reasonable system dimension.

7.2 Basic concepts on the algorithm efficiency

Knowing an algorithm for parallel computing, we note:

T_p - the execution time of an implementation of the algorithm using p processors;

T_1 - the execution time of the same implementation of the algorithm using only one processor;

T_0 - the execution time of the best implementation of the most closed sequential algorithm.

For the given parallel algoritms the efficency may be computed in many ways:

E_{num} - the numerical efficency of the parallel algorithm is computed by the following formula

$$E_{num} = \frac{T_0}{T_1}. \tag{7.1}$$

E_p^{par} - the efficency of the parallel implementation of the algorithm is given by the next realationship

$$E_p^{par} = \frac{T_1}{pT_p}.$$

E_p - the efficiency of the parallel algoritm with p processors is given by the formula

$$E_p = \frac{T_0}{pT_p} = E_{num} E_p^{par}.$$

The ideal values of these efficiency measurements are 1. The practical values are lowest because:

- for E_{num}, almost all sequential algorithms are very difficult to divide in a number of equal units (in the sense of the same computing effort);
- for E_p^{par}, the communication time between the processors is greater that the computing time necessary for an arithmetic operation. The ratio is between 500 and 1000, depending on the network;
- for E_p, since $E_{num} \leq 1$ and $E_{par}^p \leq 1$.

The notion of the speedup, S_p, is related to the second measurement:

$$S_p = \frac{T_1}{T_p}.$$

Naturally, the maximum value of S_p is p.

The employing factor of a specified processor, U, is the fraction of the execution time when the processor is busy.

7.3 First algorithm

7.3.1 Problem to be solved

We consider a stiff system in the general form

$$y'(t) = f(t, y(t)), \qquad t \in [0, T],$$

with the initial condition $y(0) = y^0$. Note with D the dimension of this system. The integration interval is divided in N equal subintervals of H lenght.

7.3.2 Basic method

We consider the extrapolation method proposed by DEUFLHARD [35], based on the linearly implicit Euler rule:

$$(I - hJ)(y_{i+1} - y_i) = hf(t_i, y_i), \qquad i \geq 0,$$

where J is an approximation to $(\partial f/\partial y)(t_0, y(t_0))$ and $y_0 = y(t_0)$. We take a sequence of integer numbers, representing some step numbers, $n_1 < n_2 < \cdots$, respectively by $n_i = i,\ i \geq 1$. We define

$$T_{j0} = y_{h_j}(t_0 + H),$$

the numerical solution obtained by performing n_j steps with size $h_j = H/n_j$. The values are extrapolated according to

$$T_{jk} = T_{j\,k-1} + \frac{T_{j\,k-1} - T_{j-1\,k-1}}{n_j/n_{j-k} - 1}, \qquad k \geq 1,$$

so that T_{jk} represents a value given by a method of $k+1$ order. To the next step, we take

$$y(t_0 + H) \approx T_{jj}$$

and $t_0 \to t_0 + H$.

DEUFLHARD [35] proves that T_{jj} is $A(\alpha)$-stable, at least for $j \leq 8$. Thus, the method has stability properties requested by the stiff systems.

7.3.3 Spliting the computational effort

The points were the approximate solution must be computed are:

$$\begin{array}{lllll}
\text{for} & y_{n_1}, & 1 \text{ point}: & t_i + H & \\
\text{for} & y_{n_2}, & 2 \text{ points}: & t_i + \frac{H}{2} \quad , & t_i + H \\
\ldots & & & & \\
\text{for} & y_{n_j} & j \text{ points}: & t_i + \frac{H}{j} \quad , \quad t_i + \frac{2H}{j} \quad , \quad \ldots \quad , & t_i + H
\end{array}$$

The number of those points is $j(j+1)/2$.

Proposition 7.1. *The computing effort can be uniformely distributed between at most* $[(j+1)/2]$ *processors, without communications.*

Proof. Note that one processor must evaluates the points of the last line. When it is not any restriction of the communication, the number of processor must divides the value $j(j+1)/2$.□

Examples:

$$j = 2 \quad \begin{cases} \text{Processor 0:} & t_i + H \\ \text{Processor 1:} & t_i + \frac{H}{2} \quad , \quad t_i + H \end{cases}$$

$$j = 4 \quad \begin{cases} \text{Processor 0:} & t_i + H \\ \text{Processor 1:} & t_i + \frac{H}{2} \quad , \quad t_i + H \\ \text{Processor 1:} & t_i + \frac{H}{3} \quad , \quad t_i + \frac{2H}{3} \quad , \quad t_i + H \\ \text{Processor 0:} & t_i + \frac{H}{4} \quad , \quad t_i + \frac{2H}{4} \quad , \quad t_i + \frac{3H}{4} \quad , \quad t_i + H \end{cases}$$

$$j = 5 \quad \begin{cases} \text{Proc. 0:} & t_i + H \\ \text{Proc. 1:} & t_i + \frac{H}{2} \quad , \quad t_i + H \\ \text{Proc. 1:} & t_i + \frac{H}{3} \quad , \quad t_i + \frac{2H}{3} \quad , \quad t_i + H \\ \text{Proc. 0:} & t_i + \frac{H}{4} \quad , \quad t_i + \frac{2H}{4} \quad , \quad t_i + \frac{3H}{4} \quad , \quad t_i + H \\ \text{Proc. 2:} & t_i + \frac{H}{5} \quad , \quad t_i + \frac{2H}{5} \quad , \quad t_i + \frac{3H}{5} \quad , \quad t_i + \frac{4H}{5} \quad , \quad t_i + H \end{cases}$$

7.3.4 Outline the algorithm

We propose the following algorithm:

Step 0. Find the number of processors, P, according to the requested approximation order. Set $m = 1$.

Step m. Each processor p (with $p = 0, \ldots, P$) follows the substeps:

(a) find the approximations in the points for which is responsable;

(b) send the computed dates and receive dates from all the other processors;

(c) find the final solution of the step, according the relation (7.1).

(d) $m \to m+1$. If $m \leq N$, continue with the next step, else stop the algorithm.

Remarks.

1. The maximum order obtained with P processors is $2P+1$. The degree of the parallelism of the proposed algorithm depends on the requested order of accuracy.

2. The approximation error produced by the parallel algorithm is the same like of the sequential algoritm. Many parallel methods proposed for the integration of differential systems does not satisfy this request.

3. The value T_{jj} can be express as a linear combination of the y_{n_j} values:

$$T_{jj} = c_1 y_{n_1} + c_2 y_{n_2} + \cdots + c_p y_{n_j}.$$

From the order conditions it results a linear system from which we get the weight values:

$$\begin{cases} c_1 + c_2 + \cdots + c_j = 1, \\ c_1 + \dfrac{c_2}{4} + \cdots + \dfrac{c_j}{j^2} = 0, \\ \vdots \\ c_1 + \dfrac{c_2}{4^{k-2}} + \cdots + \dfrac{c_j}{(j^2)^{k-2}} = 0, \\ -c_1 + c_2 + \cdots + (-1)^j c_j = 0 \end{cases}$$

The last equation is the result of the stability request at infinity. For example,

$$T_{33} = -\frac{13}{64} y_{n_1} + \frac{1}{2} y_{n_2} + \frac{45}{64} y_{n_3}.$$

Expressing T_{jj} as a linear combination of some known values, the time necessary for the stage (c) is reduced comparing with the recursive evaluation time.

7.3.5 Theoretical study of the efficiency

. Analysing the algorithm, we note that:

- The efficency of the parallel implementation of the algorithm depends on the ratio $(T_a + T_b)/T_c$, where T_x is the time necessary to acomplish the phase x $(x = a, b, c)$ at some step.

- The numerical efficiency depends on the ratio between the computing times for the $y_{h_j}(t_0 + H)$ values and the time for computing the final extrapolated value y_1.

7.4 Second algorithm

7.4.1 Problem to be solved

We consider a stiff system of the form

$$y'(t) = A(t)y(t) + g(t), \qquad t \in [0, T],$$

with the initial condition $y(0) = y^0$. Note with D the dimension of this system. The class of such stiff systems is not empty (see (S17) and (S18) from Appendix 1). For the numerical integration, the interval is divided in N equals subintervals of h lenght.

7.4.2 The basic method

Applying the implicit Euler rule

$$y_{n+1} = y_n + hf_{n+1}, \qquad n = 0, \ldots, N-1,$$

we get

$$[I - hA(t_{n+1})]y_{n+1} = y_n + hg(t_{n+1}).$$

One step n of the sequential algorithm consists on the following:

1. Evaluate $A(t_{n+1})$ and $g(t_{n+1})$.
2. Solve the linear system. If we use a GAUSS like procedure, the principal phases in solving a linear system $Qx = b$ are:
   (a) Transform Q to an upper triangular form.
   (b) Transform b in the same manner.
   (c) Solve the upper triangular linear system in x.

Remarks. The phases (1) and (2a) are independently on the initial value.

7.4.3 Splitting the computational effort

We dispose of a system with P processors, connected in a circular network. We choose two integer values K and r so that

$$KPr = N = T/h,$$

and we make the distribution of the points of the integration interval as in Figure 7.1.

The idea is to compute at each k stage, (1) and (2a) in parallel, and (2b) and (2c) in serial mode.

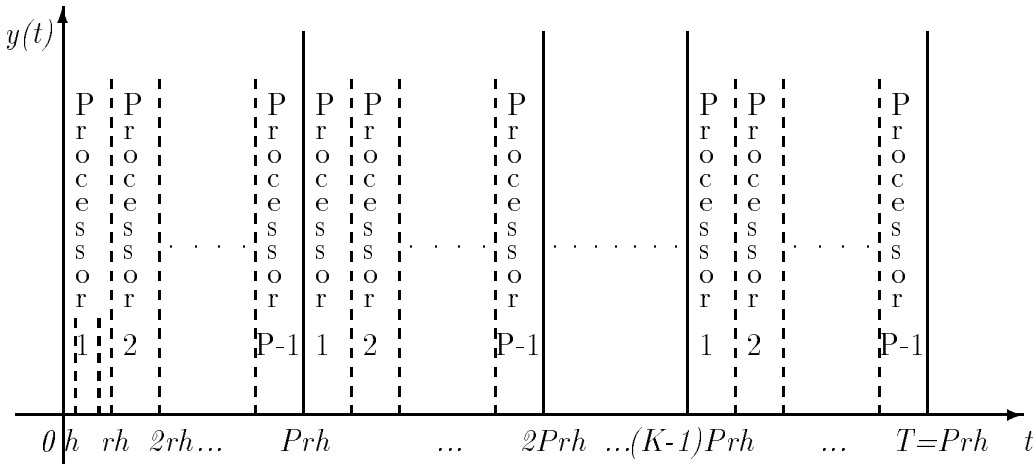

Figure 7.1: Responsability of the processors

7.4.4 Outline of the algorithm

There are K stages. In stage k, where $k = 0, \ldots, K-1$, the processor p ($p = 0, \ldots, P-1$) must execute the following:

1. for $j = 1, \ldots, r$
   (a) evaluate $A(t_{krPh+prh+jh})$ and $g(t_{krPh+prh+jh})$;
   (b) find the matrix C_j so that $C_j[I - hA(t_{krPh+prh+jh})]$ has a triangular upper form;
2. if $p \neq 0$ or $p = 0$ and $k \neq 0$, receive from processor $(p-1) \bmod P$ the vector y_{krP+pr}, else the entry vector is y_0;
3. for $j = 1, \ldots, r$
   (a) transform the input vector applying C_j;
   (b) find the solution of the transformed system, $y_{krP+pr+j}$;
4. if $p \neq P-1$ or $p = K-1$ and $k \neq K-1$, send $y_{krP+(p+1)r}$ to the processor $p+1$ and set $k \to k+1$, else stop the algorithm.

7.4.5 Theoretical study of the efficiency

Note:

F - the medium time necessary to evaluate $A(\cdot)$ and $g(\cdot)$;

U - the medium time necessary to determine a transforming matrix C;

V - the medium time necessary to transform a D-dimensional vector according to the matrix C and to find the solution of the transformed upper triangular system;

S - the medium time necessary to send a D-dimensional vector to a processor directly linked with the current processor;

R - the medium time necessary to receive a D-dimensional vector from a processor directly linked with the current processor;

G - the medium time necessary to solve a linear system of dimension D with the standard GAUSS procedure.

The time evolution of the first algorithm stage looks like in Figure 7.2. With a continue line we have plotted the time when a processor is busy.

The condition that the processor 0 does not wait until the processor $P-1$ finishis the first stage is

$$r(U+F)+rV+S+r(U+F) \geq r(U+F)+(P-1)(rV+S)+(P-2)R+R+rV+S$$

that means

$$P-1 \leq \frac{U+F}{V+(S+R)/r},$$

or

$$r \geq \frac{(S+R)(P-1)}{U+F-(P-1)V}, \quad U+F \geq (P-1)V.$$

If these inequalities are satisfied, then there are not waiting times in the stage $k>0$ (Figure 7.3).

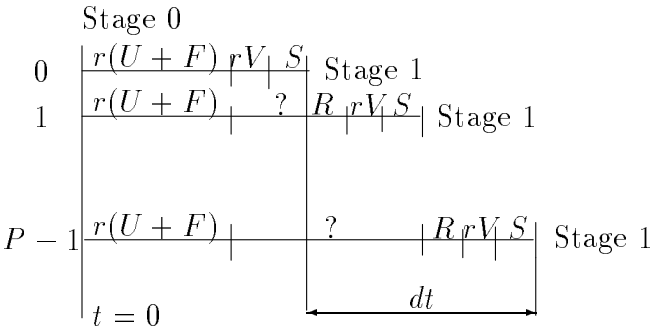

Figure 7.2: The first stage, in time

Theoretically, we have the following equations [111]:

$$T_p = K[r(U+F)+rV+S+R]+(P-1)(rV+S)+(P-2)S,$$
$$T_1 = PK[r(U+F)+rV] = PKr(U+F+V),$$

and then

$$E_p^{par} = \frac{T_1}{PT_p} = \frac{1}{1+\dfrac{K(R+S)+(P-1)(rV+S)+(P-2)S}{Kr(U+F+V)}}.$$

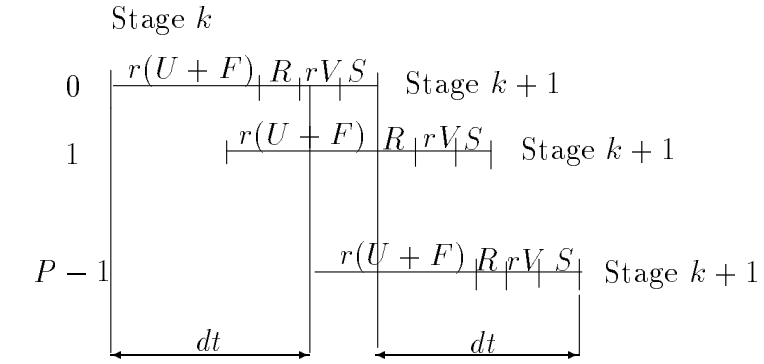

Figure 7.3: The stage $k > 0$, in time

Note that

$$T_1 = T_0 + N(U + V - G).$$

Remarks.

1. $S \approx R$ and S does not depend on D;
2. U, F and V depend superlinear on the system dimension. In addition, F depends on the system function complexity.
3. The stages number K has influence on the efficiency. For system with a small D, K must be small, because $U + F \approx S$.
4. The approximation error produced by the parallel algorithm is equal to the one produced by the sequential algorithm.
5. To maximize E_p^{par}, for a given system (for which we know the values S, T, U, F), and a given number of integration steps, N, we search (P, K, r) for which

$$\min_{(K,P,r)\in A} \frac{K(S+R) + (P-1)(rV+S) + (P-2)R}{Kr(U+F+V)},$$

where

$$A = \{(K, P, r) \in N^3 \,|\, r[U + F - (P-1)V] \geq (S+R)(P-1),\ PKr = N\}$$

For a gived P value, the optimal parameters are the followings:

$$r_{min} = \sqrt{\frac{S+R}{V}}\sqrt{N}\frac{1}{\sqrt{P(P-1)}}, \qquad K_{min} = \sqrt{\frac{V}{S+R}}\sqrt{N}\sqrt{\frac{P-1}{P}}.$$

6. For simple system function, $F \ll U$, $F \ll V$, and $U/V \leq (D+1)/4$. The condition that the processor 0 has not waiting times implies that

$$p \leq \frac{D+3}{2}.$$

Chapter 8

NUMERICAL RESULTS

The purpose of this chapter is to emphasize the implementation performances of the methods proposed in the last chapters.

The efficiency of a method can be measured in many ways. Between the computer independent measurements, we mention the number of function evaluations, the number of JACOBIAN's matrix evaluations and the number of matrix inversions. Between the computer dependent mesurements we mention the execution time of some codes associated with the methods.

8.1 Split ADAMS-MOULTON schemes

In this section we analyse the followings:

- the performances of the split ADAMAS-MOULTON scheme of second order, compared with those of the classical ADAMS-MOULTON's formula with two steps, which supposes a reduced computing effort, but it is unstable for reasonable stepsizes in the case of the integration of a stiff system;
- the performances of the split scheme with three steps, compared with the split backward differentiation formula with three steps, which supposse the same computational effort.

Similar tests are presented in [99], [112] and [114] (the test systems for which the integration with the split schemes is succesfully are (S6), (S17) and (S21) from the Appendix 1). The split methods are compared with an extrapolation method.

8.1.1 Experiment 1. Comparison between the split scheme and the classical formula

The *test systems* are the followings:

1. the linear system (S2) from the Appendix 1 integrated on the interval [0,100], with the stepsize $h = 0.5$, and starting from $y(0) = (1, 0)^T$. The ratio of the extreme eigenvalues is $S = 10^3$;

2. the nonlinear system (S20) from the Appendix 1 integrated on the interval [0,20], with the stepsize $h = 0.2$, and starting from $y(0) = (1, 1, 1, 1)^T$.

The *numerical methods* are the followings:

AM_2: the ADAMS-MOULTON's formula with two steps. The starting value for the NEWTON's iterations for solving the implicit equation is given by the ADAMS-BASHFORTH's explicit two step formula;

SAM_2: the split ADAMS-MOULTON scheme with two steps (5.7) and with $\theta = -1/3$ (for this value the predictor formula is A-stable). The scheme is A-stable. The starting value for the NEWTON's iterations is the same like for the classical method.

EX: for a high approximation of the nonlinear system solution we use the DEUHLHARD's four order extrapolation method based on the linear-implicit EULER rule (3.21), which is A(89.77°)-stable.

The *starting values* for the two steps formulae are the followings:

1. in the case of the linear system, the exact values of the solution,

2. in the case of the nonlinear system, the approximate values produced by the extrapolation method.

The NEWTON's iterations are simplified, i.e. we use only one JACOBIAN's matrix inversion, the evaluated one for the starting value.

The *results* are the followings:

1. for the first system, we draw the conclusions that

   (a) the AM_2 formula is unstable in both solution components;

   (b) the SAM_2 scheme successfully integrates the system on the given interval.

   Note the absolute error $e = (e_1, e_2, \ldots, e_M)^T$ where

   $$e_j(t) = |y_j(t) - (y_{t/h})_j|, \qquad t \in \{t_i | i = 1, \ldots, N\}, \quad j = 1, \ldots, M.$$

   For example, in Figure 8.1 we have plotted the linear interpolation curves of the discrete values to sustain the above statements.

   The computational effort was measured. At one step we have count:

   AM_2: one JACOBIAN's matrix evaluation, one matrix inversion and two function evaluations;

   SAM_2: one JACOBIAN's matrix evaluation, one matrix inversion and three function evaluations.

2. for the second system, we draw the conclusions that

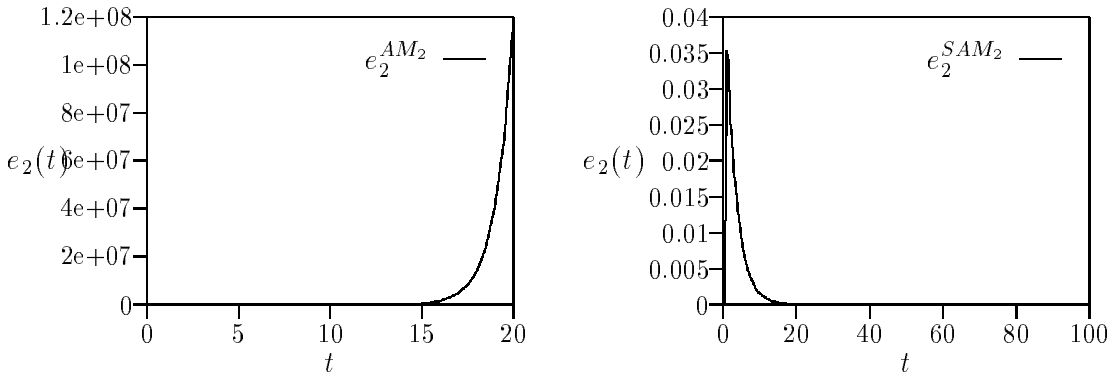

Figure 8.1: Error behaviour in the second solution component

(a) the AM_2 formula is unstable in the last two solution components (stiff components);

(b) the SAM_2 scheme successfully integrates the system.

For example, in Table 8.1 we present some discrete values of the solutions.

Applying the split scheme, for a precision $\varepsilon = 10^{-15}$ of the NEWTON's simplified iterations, there are necessary at each step: one JACOBIAN's matrix evaluation, one matrix inversion and nine function evaluations (in mean).

8.1.2 Experiment 2. Comparison between the split ADAMS-MOULTON scheme and a split backward differentiation scheme

The *test systems* are the followings:

1. the linear system (S1) from the Appendix 1 integrated on the interval [0,20], with the stepsize $h = 0.05$, and starting from $y(0) = (1,1,1,1)^T$. The ratio of the extreme eigenvalues is $S = 2 \cdot 10^2$;

2. the nonlinear system (S22) from the Appendix 1 integrated on the interval [0,0.6], with the stepsize $h = 0.002$, and starting from $y(0) = (1,0,0)^T$. The JACOBIAN's matrix eigenvalues depend on t: $\lambda_1(0) = \lambda_2(0) = 0$, $\lambda_3(t) = -0.04$, $\lambda_1(40) = 0$, $\lambda_2(40) = -3100$, $\lambda_3(40) = -0.03$;

3. the nonlinear system (S29) from the Appendix 1 integrated on the interval [0,150], with the stepsize $h = 0.5$, and starting from $y(0) = (1,1,0)^T$. The stiff ratio depends on t. For the starting values $S(0) = 1.375 \cdot 10^5$.

The *numerical methods* are the followings:

Table 8.1: Approximate solutions of the nonlinear system

t	t/h	$y(t) \approx y_{t/h}^{EX}$	$y_{t/h}^{SAM_2}$	$y_{t/h}^{AM_2}$
2	10	$\begin{pmatrix} 1.856903 \\ 0.033898 \\ 0.034356 \\ 0.034451 \end{pmatrix}$	$\begin{pmatrix} 1.864032 \\ 0.034188 \\ 0.0344793 \\ 0.033156 \end{pmatrix}$	$\begin{pmatrix} 1.863837 \\ 0.034183 \\ 0.373462 \\ 3.131642 \end{pmatrix}$
10	50	$\begin{pmatrix} 1.999940 \\ 0.0399973 \\ 0.040014 \\ 0.040030 \end{pmatrix}$	$\begin{pmatrix} 1.999955 \\ 0.039998 \\ 0.040014 \\ 0.040030 \end{pmatrix}$	$\begin{pmatrix} 1.999954 \\ 0.039998 \\ 54.65505 \\ 7682697.13 \end{pmatrix}$
20	100	$\begin{pmatrix} 2 \\ 0.04 \\ 0.040016 \\ 0.040032 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 0.04 \\ 0.040016 \\ 0.040032 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 0.04 \\ 31365.890 \\ 7.568 \cdot 10^{14} \end{pmatrix}$

SAM_3: the split ADAMS-MOULTON scheme with three steps and of four order (5.9) with $\theta = -7/24$, in which case the scheme is A(89.296°)-stable. The starting value for the NEWTON's simplified iterations are produced by the ADAMS-BASHFORTH's three-step explicit formula;

$SBDF_3$: the CASH's split backward differentiation scheme (2.35) with three steps and third order (which supposes a similar effort like the above method) for $\theta = -1.$, in which case the scheme is L-stable. The starting value for the NEWTON's simplified iterations are produced by the EULER's explicit rule;

RK: the standard RUNGE-KUTTA's process of four order, implemented for finding the stiff components of the systems;

EX: the analougous extrapolation method to that of the first experiment, but of five order, for a high approximation of the exact solution for comparing the above methods.

The *starting values* for the two steps formulae are the followings:

1. in the case of the linear system, the exact values of the solution;

2. in the case of the nonlinear systems, the approximate values produced by the extrapolation method.

Like in the first experiment, the NEWTON's procedure is implemented in a simplified version with only one matrix inversion. The tolerated error between two succesive iterations is $\varepsilon = 10^{-9}$.

The *results* are the followings:

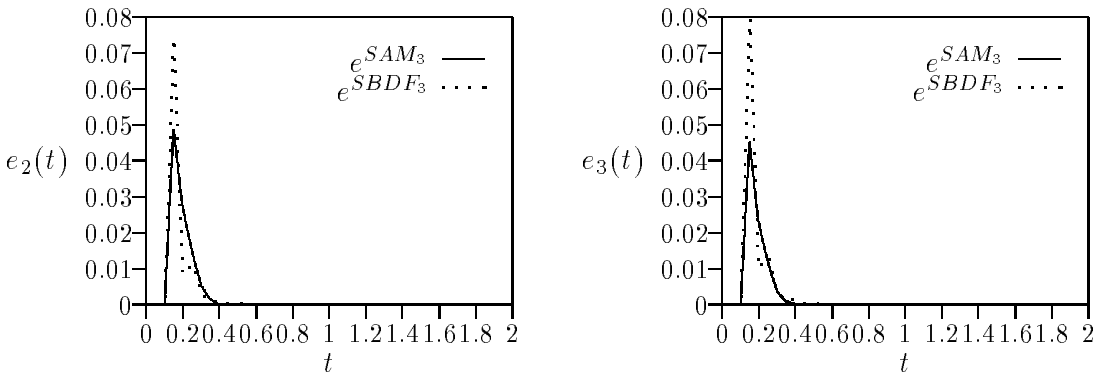

Figure 8.2: Absolute errors in the first system stiff components

1. for the first system, we draw the conclusions that

   (a) the RK process can not integrate the stiff components, y_3, y_4, though it gives satisfactory results in the first two components;

   (b) the split schemes successfully integrate the system;

   (c) the error produced by the approximation of the exact solution with the SAM_3 scheme is lowest than the one produced by the $SBDF_3$;

   (d) the computational effort is the same for both split schemes: three function evaluations, one JACOBIAN's matrix evaluation, and one matrix inversion.

   For example, in Figure 8.2 we have plotted the absolute error curves.

2. integrating the second system, the RK process gives an unstable solution (approximate values for only 12 steps). The stiff component is y_2. In Figure 8.3 we have plotted the approximate solutions produced applying the two split schemes and the absolute error curves.

3. for the third system, we draw the conclusions that:

   (a) the solution produced by the RK process is unstable in all solution components;

   (b) the error produced by the split CASH's scheme is unbounded;

   (c) the split ADAMS-MOULTON scheme successfully integrates the system and the mean computing effort at one step can be express in nine function evaluations, one matrix inversion, and one JACOBIAN's matrix evaluation.

   For example, in Table 8.2, we present some significant values.

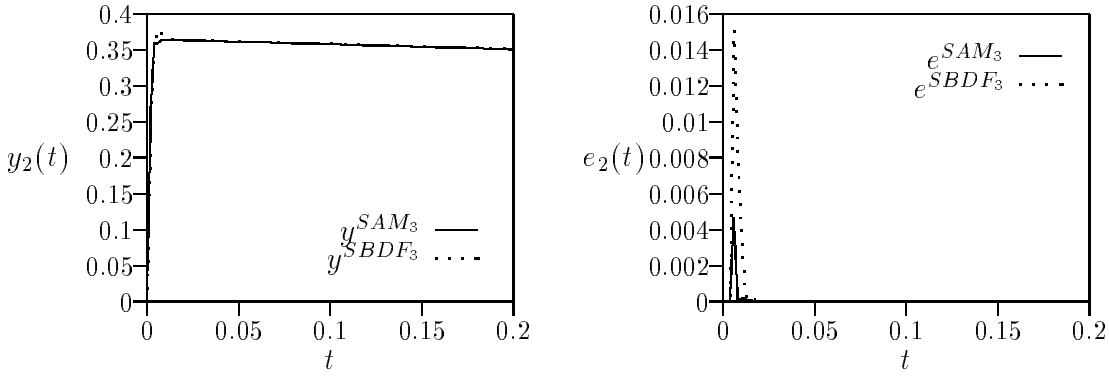

Figure 8.3: Approximate solutions and absolute error in the second component of the exact solution

For the instability of the other schemes, we can mention some values:

$$y_4^{BDFS_3} = \begin{pmatrix} 7668.407 \\ 322.1616 \\ 423.4813 \end{pmatrix}, \qquad y_{10}^{BDFS_3} = \begin{pmatrix} 279679399.97 \\ 11947028.66 \\ 15687767.34 \end{pmatrix},$$

$$y_1^{RK} = \begin{pmatrix} -3673 \\ 6.413 \\ 6.936 \end{pmatrix}, \qquad y_3^{RK} = \begin{pmatrix} 7.044 \cdot 10^{66} \\ 1.525 \cdot 10^{40} \\ 1.943 \cdot 10^{40} \end{pmatrix}.$$

8.1.3 Conclusions

The split ADAMS-MOULTON formulae can be used with success in the integration of stiff systems. Altrough using a such scheme is more expensive in the sense of the computing effort, the advantage is the possibility to integrate some systems for which the classical ADAMS-MOULTON's formulae reach some stability difficulties.

Comparing the split ADAMS-MOULTON scheme with the CASH's split backward differentiation scheme with the same stepnumber, one can see that the first one produces a discrete solution more closed to the exact solution than that produced by the second one.

The split scheme can be implemented in a numerical code which:

1. uses an ADAMS-MOULTON's formula as basic formula;
2. includes an instability test which determines the situation when we deal with a stiff system;
3. changes the basic formula with the split corresponding scheme when the instability test is successfully.

Table 8.2: Approximate solution produced by SAM_3 integrating the third system

t	t/h	$y(t) \approx y_{t/h}^{EX}$	$y_{t/h}^{SAM_3}$	$e(t) \approx \lvert y_{t/h}^{EX} - y_{t/h}^{SAM_3} \rvert$
5	10	$\begin{pmatrix} 1.251492 \\ 1.070882 \\ 0.608467 \end{pmatrix}$	$\begin{pmatrix} 1.254754 \\ 1.070882 \\ 0.608069 \end{pmatrix}$	$\begin{pmatrix} 0.003262 \\ 6.164 \cdot 10^{-7} \\ 0.000399 \end{pmatrix}$
50	100	$\begin{pmatrix} 1.625682 \\ 1.558227 \\ 7.302124 \end{pmatrix}$	$\begin{pmatrix} 1.626610 \\ 1.559085 \\ 7.300877 \end{pmatrix}$	$\begin{pmatrix} 0.000927 \\ 0.000858 \\ 0.001247 \end{pmatrix}$
150	300	$\begin{pmatrix} 0.819860 \\ 0.950671 \\ 20.38286 \end{pmatrix}$	$\begin{pmatrix} 0.820188 \\ 0.951237 \\ 20.39756 \end{pmatrix}$	$\begin{pmatrix} 0.000328 \\ 0.000565 \\ 0.014700 \end{pmatrix}$

A such implementation version has an important advantage: the stored values from one step to another must be not modified when we change the formula.

8.2 Split second derivative scheme

In this section we analyse the followings:

- the performances of the split ENRIGHT scheme with three steps (5.17), compared with those of the classical ENRIGHT's second derivative formula with three steps (5.2), which has some difficulties to integrate the stiff systems with some eigenvalues closed to the imaginary axis;
- the performances of the split second derivative backward differentiation scheme with four steps (5.20) compared with those of the classical second derivative backward differentiation formula with the same stepnumber (3.3) which is only stiffly stable.

For the chosen θ values, the split schemes are A-stable.

Similar tests are presented in [109].

8.2.1 Experiment 1. Classical formula instability

The *test system* is a modification of the system (S9) from the Appendix 1:

$$\begin{cases} y_1'(t) = ay_1(t) + by_2(t), \\ y_2'(t) = -by_1(t) + ay_2(t) \end{cases}$$

The initial values are $y(0) = (0, 1)^T$, and the integration interval $[0, T]$. The exact solution is $y(t) = (e^{at} \sin bt, e^{at} \cos bt)^T$.

We propose the following values:

1. for the split ENRIGHT scheme, $h = 0.5,\ a = -0.04,\ b = 3.6,\ T = 50$;

2. for the split second derivative backward differentiation scheme, $h = 0.5,\ a = -0.05,\ b = 12,\ T = 10$.

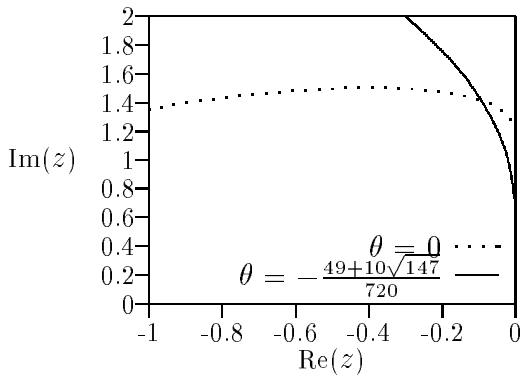

Figure 8.4: Exact and approximate solutions produced by EN_3 and SES_3 in the first component

The *numerical methods* are the folowings:

EN_3: the ENRIGHT's second derivative formula with three steps and of five order (5.2). The starting value for the simplified NEWTON's iterations, for solving the implicit predictor formula, is the one produced by the EULER's explicit rule.

SES_3: the split ENRIGHT scheme with three steps and of five order (5.17) with $\theta = 1/12$ (for this value the scheme is A-stable). The starting value is the same like for the above mentionated method.

$SBDF_4$: the second derivative backward differentiation formula with four steps and of five order (5.3).

$SSBDS_4$: the split second derivative backward differentiation scheme, with four steps and of five order, (5.20), with $\theta = -1/3$ (for this value the scheme is A-stable).

As starting values for the multistep formulae we have take the exact values of the solution.

The NEWTON's procedure for solving the implicit equations is built-up in a simplified variant with only one matrix inversion (the JACOBIAN's matrix evaluated at the starting value).

We have get the following results:

1. the EN_3 and $SBDF_4$ formulae are unstable in the both solution components;

2. the SES_3 and $SSBDS_4$ schemes successfully integrate the system.

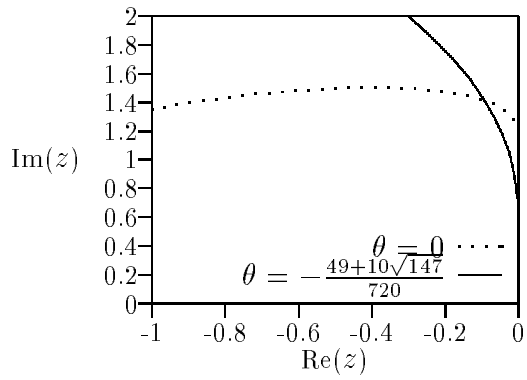

Figure 8.5: Exact and approximate solutions produced by $SBDF_4$ and $SSBDS_4$ in the second component

For example, in Figures 8.4 and 8.5 we have plotted the exact solution and the approximate ones.

The computational effort to one step is the following:

- for EN_3 and $SBDF_4$, one JACOBIAN's matrix evaluation, one matrix inversion, two function evaluations and one derivative evaluation;

- for SES_3 and $SSBDS_4$, one JACOBIAN's matrix evaluation, one matrix inversion, three function evaluations and two derivative evaluations.

8.2.2 Experiment 2. Error of the split schemes

The *test systems* are the followings:

1. the linear system (S2) from the Appendix 1 integrated on the interval [0,20], with the stepsize $h = 0.5$, and starting from $y(0) = (1, 0)^T$;

2. the nonlinear system (S29) from the Appendix 1 integrated on the interval [0,100], with the stepsize $h = 0.5$, and starting from $y(0) = (1, 1, 0)^T$.

The *numerical methods* are the same like in the first experiment.
We have get the following *results*:

1. the errors (absolute or relative) produced by the approximation of the exact solution with the split schemes are lowest than the ones produced by the classical formulae;

2. the relative error produced by $SSBDS_4$ is lowest than the one of the SES_3.

To prove these statements, in Figure 8.6 we have plotted the absolute error curves for the first system, and in Figure 8.7, the relative error curves for the second system.

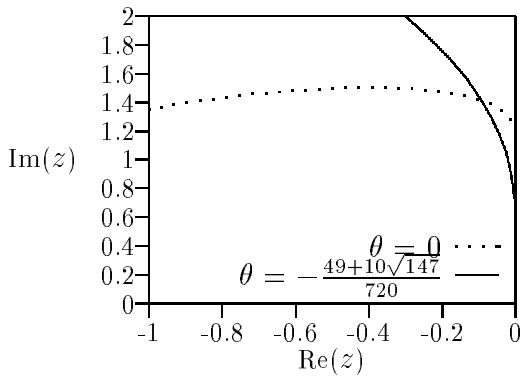

Figure 8.6: Absolute errors approximating the first system solution

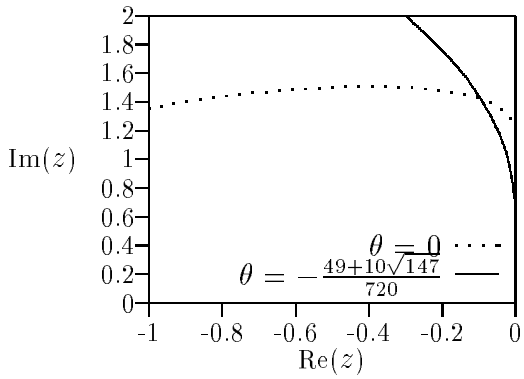

Figure 8.7: Relative errors approximating the second system solution

8.2.3 Conclusions

The results of the first experiment confirm that the stability domains of the split schemes are largest than the ones of the classical corresponding formulae.

The split schemes produce some approximation errors lowest than the ones corresponding to the classical formulae. Comparing the split ENRIGHT scheme with the split second derivative backward differentiation scheme with the same order (but with different stepnumbers), we can see that, for the second one, the error has some smallest discrete values.

The proposed split schemes are proving their efficiency in some implementations like the following:

1. use a second derivative multistep formula as basic formula;

2. include an instability test which identifies the situation when the stiff system

does not correspond to the stability domain of the basic formula;

3. switch from the basic formula to the split corresponding scheme if the instability test is successfully.

8.3 Exponentialy fitted formulae

In this section we study the performances of the parameter dependent backward differentiation formula with two steps, compared with two GEAR's backward differentiation formulae, one of the same order, the other of the same stepnumber.

Applying the method of exponential fitting, we expect that the parameter dependent formula produces the smallest error.

Similar test are presented in [100]. We must mention the very good results in the integration of the linear system (S12) and the nonlinear system (S23).

8.3.1 Experiment 1. Backward differentiation formulae applied to a nonhomogeneous equation

The *test equation* is (S10) from the Appendice 1, which is integrated on the interval [0,100], with the stepsize $h = 0.1$, and starting from $y(0) = 10$. The exact solution is $y(t) = E(t) + 10e^{-200t}$.

Note the following *numerical methods*:

BDF_1: the GEAR's onestep backward differentiation formula (1.12) of first order (the EULER's implicit rule) which is A-stable;

BDF_2: the GEAR's two-step backward differentiation formula (2.31) of second order which is also A-stable. The starting value for the multistep formula is the exact value of the solution.

BDF_2^q: the parameter dependent backward differentiation formula (6.3) with two steps and of first order which is also A-stable for an a value chosen by the request of exponential fitting:

$$a \equiv a(q) = \frac{(2q-3)e^{2q} + 4e^q - 1}{(e^q - 1)^2},$$

where

$$q \equiv q(t) = \begin{cases} -200, & e^{-200t} > \varepsilon, \\ -1, & \text{altfel}, \end{cases}$$

The switch between the two fitting points is made when the stiff component, $10e^{-200t}$, no more affects the solution: ε is chosed so that $10\varepsilon < 10^{-15}$.

EU: the EULER's explicit rule for proving the stiff character of the system.

We draw the *conclusion* that, for the test equation, the exponential fitted formula produces an approximation error lowest than that produced by the two GEAR's formulae. The unstable behaviour of the EU's solution lie to the conclusion that we deal with an stiff equation.

In Figure 8.8 one can see the plots of the absolute error curves.

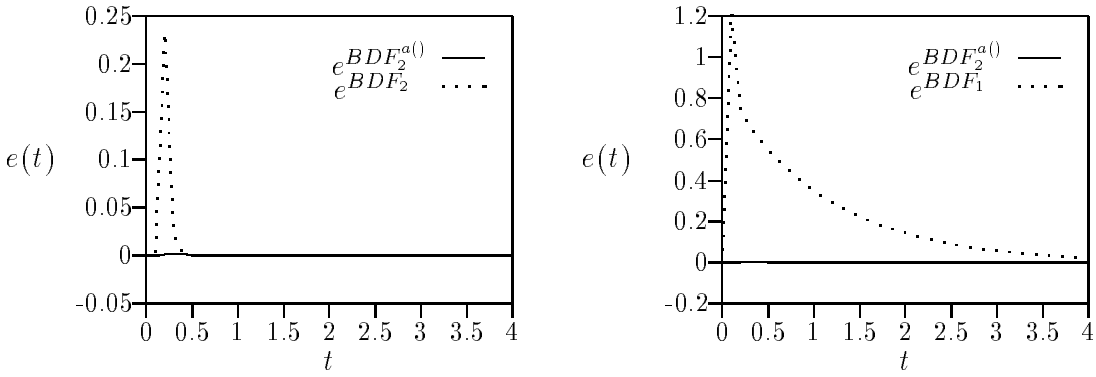

Figure 8.8: Approximation errors produced by the backward differentiation formulae

8.3.2 Experiment 2. Backward differentiation formulae applied to a perturbated linear system

We have tested the methods from the first experiment in the integration of the system (S13) from the Appendice 1, on the interval [0,25], with the stepsize $h = 0.02$. The eigenvalues ratio is $S = 1.5 \cdot 10^3$.

The a parameter is chosed as

$$a \equiv a(q) = \frac{(2q-3)e^{2q} + 4e^q - 1}{(e^q - 1)^2},$$

where $q = -h$.

We draw the *conclusion* that, for the given system, the parameter dependent formula produces a lowest error than the one of the GEAR's formulae. The EU formula has an unstable behaviour for the given stepsize. For proving this statement we have plotted in Figure 8.9 the absolute error curves in each solution component.

8.3.3 Experiment 3. Backward differentiation formulae applied to a nonlinear system

We draw the same *conclusion* like for the last two experiments, when we integration the nonlinear system (S20) from the Appendice 1, in the interval [0,20], with the stepsize $h = 0.2$, and the starting values $y(0) = (1, 1, 1, 1)^T$.

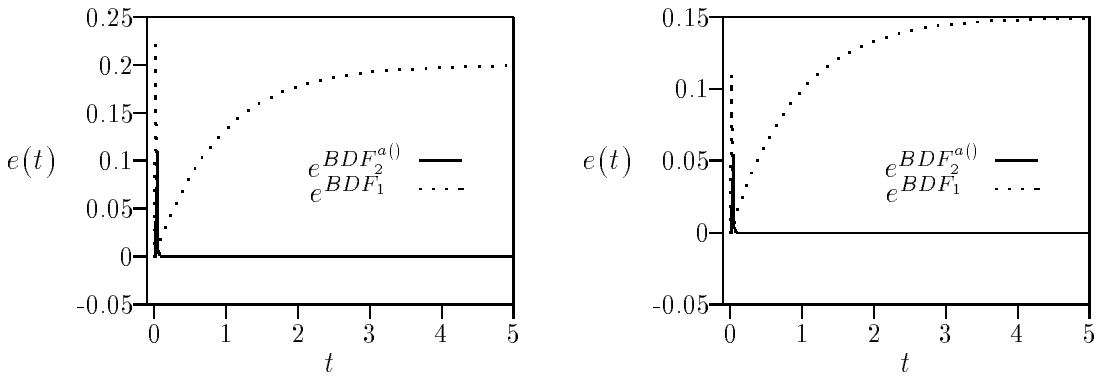

Figure 8.9: Absolute errors in the approximation of the perturbated linear system solution

The method which give us an high approximation solution is the DEUFLHARD's extrapolation method of four order, based on the linear-implicit EULER rule (3.21) (noted EX). We use also the standard RUNGE-KUTTA's process of four order for finding the stiff components of the system (noted RK).

The a parameter is chosed as

$$a \equiv a(q) = \frac{(2q-3)e^{2q} + 4e^q - 1}{(e^q - 1)^2},$$

where $q = -100 \cdot h$.

The RK method application indicates that the third and the fourth components of the exact solution are stiff components.

For proving the above conclusion we have plotted in the Figure 8.10 the absolute errors in two solution components with disctinct character (the first one is a smooth component, the second one, a stiff component).

The computing effort is the same for both two step formulae.

8.3.4 Conclusions

From the point of view of the aproximation error, an exponential fitted backward differentiation formula is more proper for the integration of a stiff system than the classical GEAR's formulae.

8.4 Nonlinear formulae

We study the efficiency in the implementation of one exponentially fitted nonlinear method presented in Chapter 5, compared with closed formulae:

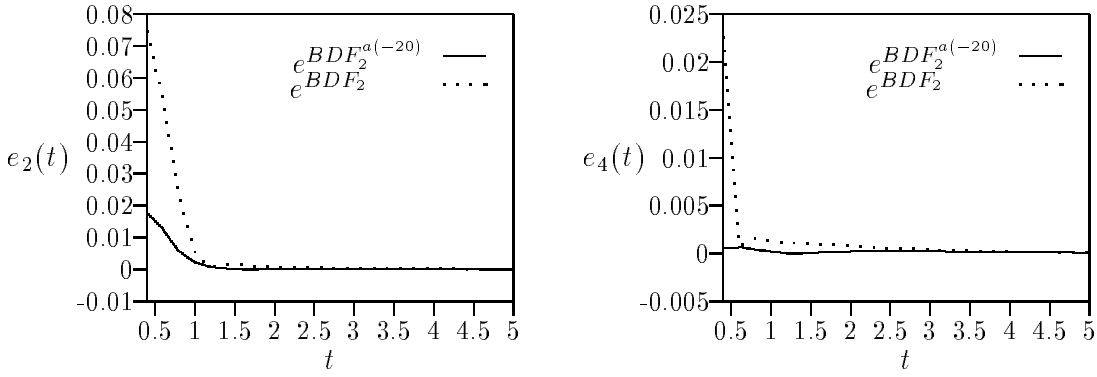

Figure 8.10: Absolute errors in the approximation of two distinct components

- the linear-implicit EULER's rule;
- the linear-implicit midpoint rule;
- the EULER's explicit rule.

The comparison between some A-stable nonlinear explicit formulae and some stiffly-stable linear multistep formulae can be done only from the error point of view. Comparing the computational effort, we draw the conclusion that the explicit nonlinear formulae are more convenient.

Similar tests are presented in [103] and [115]. The discussed methods are those with exponentially fitted coeficients, and the comparing methods are the trapezoidal rule, the GEAR's backward differentiation formulae and the ENRIGHT's second derivative formulae. We can mention the improvement in error decrease using the nonlinear methods in the integration of the system (S14) and (S22).

8.4.1 Experiment 1. Integrating a separable system

We consider the *test system* (S21) from the Appendix 1 on the integration interval [0,20], with the stepsize $h = 0.1$. The initial ratio of the JACOBIAN's matrix eigenvalues is $S = 10^6$.

Note the following *numerical methods*:

NL: the nonlinear method with two steps (6.18), with maximum third order, which is A-stable. The starting value is the exact solution value at $t = h$;

EL: the linear-implicit EULER's rule (3.21) of first order, which is A-stable;

EE: the EULER's explicit rule of first order, for finding the stiff components of the exact solution.

We have get the following *results*:

1. the behaviour of the EE solution lies us to the conclusion that the first two components of the exact solution are stiff components;

2. the nonlinear formula produces the exact solution.

For proving these statements, we present in Table 8.3 some values of the numerical solutions.

Table 8.3: Exact and approximate solutions of the separable system

$y(t)$	$y_{t/h}^{NL}$	$y_{t/h}^{EL}$
$y(2h) = \begin{pmatrix} 0 \\ 0 \\ -4.509 \\ -0.8332 \end{pmatrix}$	$y_2^{NL} = \begin{pmatrix} 0 \\ 0 \\ -4.509 \\ -0.8332 \end{pmatrix}$	$y_2^{EL} = \begin{pmatrix} -1.077 \cdot 10^{-4} \\ -1.675 \cdot 10^{-4} \\ -7.75 \\ -0.8455 \end{pmatrix}$
$y(7h) = \begin{pmatrix} 0 \\ 0 \\ -9.919 \\ -0.5879 \end{pmatrix}$	$y_7^{NL} = \begin{pmatrix} 0 \\ 0 \\ -9.919 \\ -0.5879 \end{pmatrix}$	$y_7^{EL} = \begin{pmatrix} -1.025 \cdot 10^{-14} \\ -4.803 \cdot 10^{-14} \\ -9.93 \\ -0.6064 \end{pmatrix}$
$y(200h) = \begin{pmatrix} 0 \\ 0 \\ -10 \\ -0.4712 \end{pmatrix}$	$y_{200}^{NL} = \begin{pmatrix} 0 \\ 0 \\ -10 \\ -0.4712 \end{pmatrix}$	$y_{200}^{EL} = \begin{pmatrix} 0 \\ 0 \\ -10 \\ -0.04783 \end{pmatrix}$

From the calculus effort point of view, we mention that, at each step, there are necessary the followings:

1. for NL, one JACOBIAN's matrix and one function evaluations;

2. for EL, one JACOBIAN's matrix and one function evaluations;

3. for EE, one function evaluation.

8.4.2 Experiment 2. Integrating a nonlinear system

The *test system* is (S20) from the Appendix 1, integrated on the interval [0,20] with the starting values $y(0) = (1, 1, 1, 1)^T$, and the stepsize $h = 0.2$. Note that the JACOBIAN's matrix eigenvalues are the same in all the integration interval. The ratio between the largest and the smallest one is $S = 10^2$. The JACOBIAN's matrix is diagonally dominated.

The *numerical methods* are the followings:

NL: the above mentionated nonlinear formula. The starting value is produced by the EX method;

EX: the DEUFLHARD's extrapolation method of four order for a high approximation of the exact solution;

PM: the linear-implicit midpoint rule, which is A-stable, and has second order;

EE: the EULER's explicit rule, for finding the stiff components of the exact solution.

We draw the following *conclusions*:

1. the EE solution is unstable in the last three components (stiff components);
2. the nonlinear formula produces the exact solution in the first component;
3. the approximation error produced applying the NL formula is lowest than the one produced by PM, in the first half of the integration interval.

The computational effort is evaluated at one step:

1. for PM are necessary one function evaluation and two multiplication between a matrix and a vector;
2. for NL are necessary one JACOBIAN's matrix evaluation and one multiplication between a matrix and a vector.

To prove the above statements, we mention in Table 8.4 the approximate values of the exact solution.

Table 8.4: Exact and approximate solutions of the nonlinear system

$y(t)$	$y_{t/h}^{NL}$	$y_{t/h}^{PM}$
$y(2h) \approx y_2^{EX} = \begin{pmatrix} 1.319 \\ 0.0666 \\ 0.1617 \\ 0.0166 \end{pmatrix}$	$y_2^{NL} = \begin{pmatrix} 1.319 \\ 0.7373 \\ 0.1934 \\ 0.0205 \end{pmatrix}$	$y_2^{PM} = \begin{pmatrix} 1.333 \\ -0.3152 \\ -0.5149 \\ -0.8743 \end{pmatrix}$
$y(10h) \approx y_{10}^{EX} = \begin{pmatrix} 1.8563 \\ 0.0335 \\ 0.3401 \\ 0.0352 \end{pmatrix}$	$y_{10}^{NL} = \begin{pmatrix} 1.8563 \\ 0.2284 \\ 0.2549 \\ 0.0270 \end{pmatrix}$	$y_{10}^{PM} = \begin{pmatrix} 1.8680 \\ 0.0303 \\ 0.0965 \\ -0.5431 \end{pmatrix}$
$y(100h) \approx y_{100}^{EX} = \begin{pmatrix} 2 \\ 0.0400 \\ 0.4002 \\ 0.0416 \end{pmatrix}$	$y_{100}^{NL} = \begin{pmatrix} 2 \\ 0.0400 \\ 0.3913 \\ 0.0405 \end{pmatrix}$	$y_{100}^{PM} = \begin{pmatrix} 2 \\ 0.0400 \\ 0.4002 \\ 0.0478 \end{pmatrix}$

8.4.3 Conclusions

The tested nonlinear formula can successfully integrates some stiff systems. The approximation error is comparable with those produced by some implicit A-stable linear multistep formulae.

8.5 Onestep hybrid methods

In this section we study the performances of two hybrid methods proposed in the Chapter 6, compared with an ENGLAND's method [37] with the same computational effort.

We mention that in the papers [98], [102] and [106] are given the test resulting from other three hybrid methods proposed in Chapter 6. The compared methods are the ENRIGHT's onestep second derivative formula, the OBRECHKOFF's second derivative formula, and a BOCKHOVEN's hybrid scheme. The test systems are (S12), (S13), (S17) and (S25) from the Appendice 1.

8.5.1 Experiment 1. Integrating one stiff equation

The *test equation* is (S10) from the Appendice 1, integrated on the interval [0,20], with the stepsize $h = 0.5$, and the starting value $y(0) = 10$. The exact solution is $y(t) = E(t) + 10e^{-200t}$.

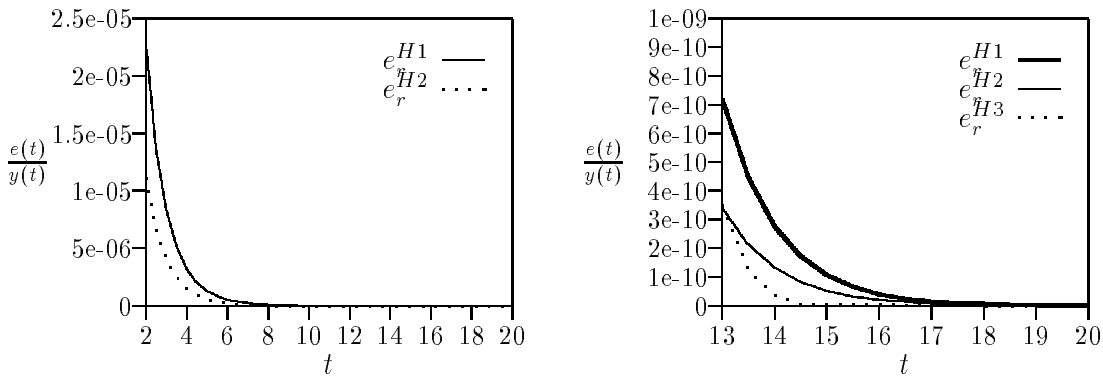

Figure 8.11: Relative errors in the integration of the stiff equation

The tested *numerical methods* are the followings:

H1: the ENGLAND's L-stabil scheme of third order (6.19);

H2: the L-stable hybrid method of third order (6.25);

H3: the strong A-stable hybrid method of third order (6.24);

RK: the standard RUNGE-KUTTA's process of four order.

The starting value for the NEWTON's procedure for solving the implicit equations is given by the EULER's explicit rule.

We draw the *conclusion* that the first three methods successfully integrate the system. The lowest error is produced by H2 in a first integration interval, and by H3 in the second one. In Figure 8.11 we have plotted the relative errors curves.

The computational effort is the same for the first three methods and consists, at each step, in two function evaluations, one JACOBIAN's matrix evaluation, and one matrix inversion.

8.5.2 Experiment 2. Integrating an linear system

We test the same *numerical methods* on the system (S2) from the Appendice 1, integrated on the interval [0,20], with the stepsize $h = 0.5$, and the starting values $y(0) = (1,0)^T$. The exact solution of the problem is $y_1(t) = 2e^{-t} - e^{-1000t}$, $y_2(t) = -e^{-t} + e^{-1000t}$.

We get the following *results*:

1. the H1 and H2 methods produce approximately the same absolute error (the small difference between them indicates that the H2 solution is more closed to the exact solution);

2. the H3 method produces a lowest error than the ones of the H1 and H2 methods only in a second integration subinterval;

3. the RK can not integrate the problem.

In Figure 8.12 one can see the plots of the relative errors curves.

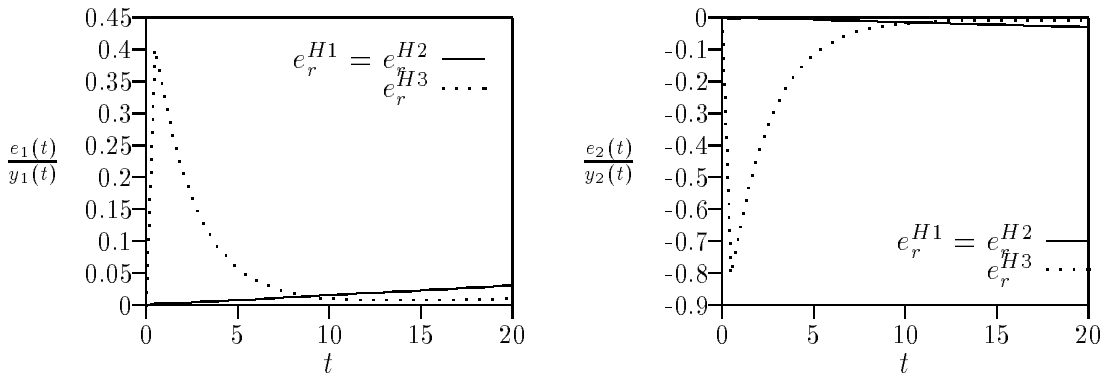

Figure 8.12: Relative errors of the approximative solutions

Table 8.5: Approximative solution in the integration of the nonlinear system

t	$y_{t/h}^{H1}$	$\lvert y_{t/h}^{EX} - y_{t/h}^{H1}\rvert$	$y_{t/h}^{H2}$	$\lvert y_{t/h}^{EX} - y_{t/h}^{H2}\rvert$	$y_{t/h}^{RK}$
0.2	1.181287	0.004596	1.181286	0.004596	1.181
	0.122202	0.096288	0.122204	0.096286	0.3427
	0.058113	0.090867	0.061166	0.087815	85.62
	−0.055171	0.064238	−0.054026	0.063094	5515
1	1.632159	0.012007	1.632159	0.012007	1.632
	0.025358	0.000739	0.025359	0.000739	0.02946
	0.263383	0.011295	0.263386	0.011298	$1.267 \cdot 10^{10}$
	0.027210	0.001211	0.027210	0.001211	$1.078 \cdot 10^{19}$
2	1.864693	0.008383	1.864693	0.008383	1.865
	0.034215	0.000694	0.034215	0.000694	0.03424
	0.346532	0.006399	0.346532	0.006399	$2.071 \cdot 10^{20}$
	0.035929	0.000699	0.035929	0.000699	$1.611 \cdot 10^{39}$

Since the system is linear, the computing effort is the same for the first three methods. At one step, there are necessary two function evaluations, one JACOBIAN's matrix evaluation, and one matrix inversion.

8.5.3 Experiment 3. Integrating a nonlinear system

We test only *H1 and H2 schemes* in the integration of a nonlinear system, in the aim to prove that the produced errors are the same or very closed each another. The *test nonlinear system* is (S20) from the Appendice 1, integrated on the interval [0,2], with the stepsize $h = 0.2$.

For a high approximative solution we use the DEUHLHARD's extrapolation method of four order.

The starting value for the NEWTON's procedure for solving the implicit equations is that produced by the EULER's explicit rule.

The *results* confirm the above statement (see Table 8.5).

There are no reported differences between H1 and H2 scheme in the computational volume.

8.5.4 Conclusions

We have numerical proved that the first third order hybrid method, proposed in Chapter 6, is comparable with the ENGLAND's hybrid method of the same order in both senses, of approximation error and computational effort.

The second scheme is proper for integration when we are intersted only in the solution at the final integration value.

8.6 Parallel algorithms

In this section we study the performances of the parallel algorithms proposed in Chapter 7. The tests was made on a transputer system (T-800) under the PARIX operation system. The results of these tests are also presented in the paper [110].

8.6.1 Experiment 1. The efficiency of the parallel extrapolation algorithm

The *test systems* are the followings:

1. a generalization of (S6) from the Appendix 1:

$$y_i'(t) = -i^5 y_i(t), \qquad t \in [0,1], \qquad i = 1, \ldots, D.$$

   For $D = 10$ we get (S6).

2. the system (S40) from the Appendix 1.

The *implemented algorithms* are the followings:

- the parallel algorithm from section 7.3.4;
- the sequential DEUFLHARD's extrapolation method (with minimum number of arithmetical operations)

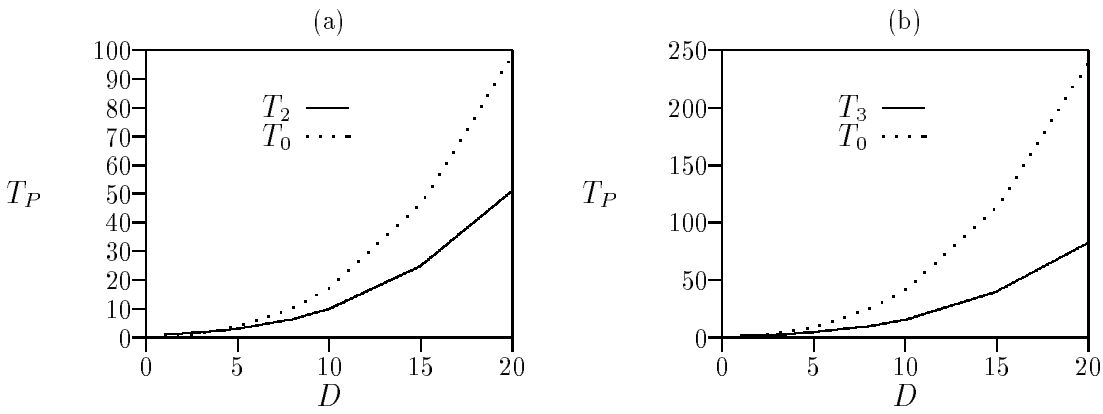

Figure 8.13: The running time of some algorithm implementations (a) for $P = 2$ processors (b) for $P = 3$ processors

We study the effect of changing:

1. the system dimension D;

2. the stepnumber N;

3. the processor number P.

We have measured the folowings times:

- the execution time of the sequential code, T_0;

- the execution time of the parallel code using P processors, T_P;

- the medium computing time of a processor, T_P^c.

We present the *results* in a graphical form.

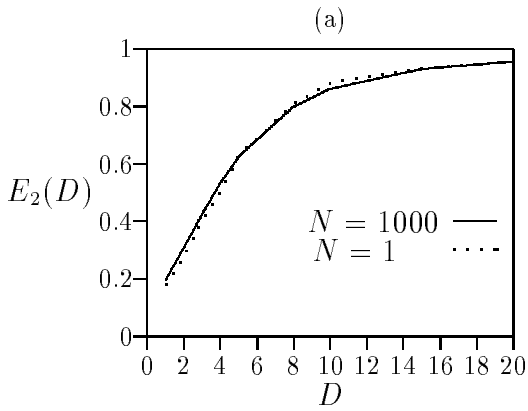

Figure 8.14: The efficiency $E_2(D, N)$ for some (a) step numbers (b) dimensions

Analysing the figures we draw the following conclusions:

1. By the implementation of the parallel algorithms, the running time of the extrapolation method can be improved (Figure 8.13).

2. The efficiency E_2 of the parallel algorithm, which use $P = 2$ processors, depends on the dimension D of the system (Figure 8.14.a), but not on the stepnumber N (Figure 8.14.b). For $D \geq 10$, E_2 is closed to the ideal value;

3. For some given values N and D, the efficiency has not a significant dependence on the number of processors, P (Figure 8.15.a). This fact is not valid for most parallel algorithms. The explanation lies, in our case, to the fact that the computing time of the sequential algorithm, T_0, depends on the requested accuracy of the numerical solution, which is a function on P.

4. For some given value N and P, the total computing time increases with the dimension D so that it covers the negative effects of the communications (Figure 8.15.b).

5. The employing factor U does not depends on the number of processors P and on the stepnumber N (Figure 8.16).

The efficiency, for fixed stepnumber, dimension and processors, depends on the complexity of the system function. For instance, for the second system, the efficiency of the parallel algorithm, with $N = 20000$ and $P = 2$ processors, is 85.191%.

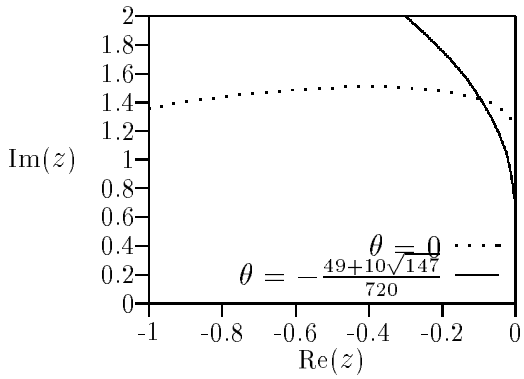

Figure 8.15: (a) The efficency $E_P(D)$ for $N = 1000$ subintervals (b) The communications influence for $N = 1000$ and $P = 3$

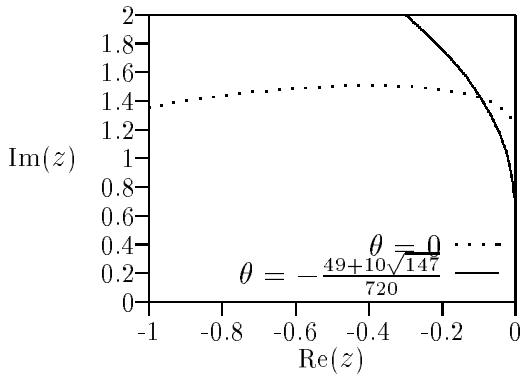

Figure 8.16: The employing factor $U(D, P)$ (a) for $N = 1000$ subintervals (b) for $P = 2$ processors

8.6.2 Experiment 2. Parallel algorithm for linear stiff systems

We have tested the proposed algorithm on the system (S17) and on the following system

$$y_i'(t) = -i^5 y_i(t), \qquad t \in [0,1], \quad i = 1, \ldots, D.$$

This system generalizes in D dimension the stiff system (S6).

The *implemented algorithms* are the followings:

- the parallel algorithm from section 7.4.4;
- the EULER's implicit rule with the GAUSS's procedure for solving the linear resulting system;

Like in the first experiment, we test the influence on the algorithm performances of the N, D and P parameters and of the system function complexity.

Analysing Table 8.6, we draw the conclusion that the efficiency of the parallel implementation of the algorithm increases with the dimension D and decreases with the processors number P. For $P = 2$, E_P^{par} is very closed to the ideal value.

Table 8.6: Efficiency in the integration of first system with $N = 1000$

	E_p^{par}			E_p			E_{num}
D\P	2	4	8	2	4	8	
2	0.962	0.500	0.260	0.683	0.355	0.185	0.710
3	0.973	0.610	0.303	0.700	0.440	0.218	0.720
4	0.978	0.667	0.331	0.703	0.476	0.236	0.714
5	0.983	0.712	0.355	0.700	0.508	0.253	0.713
8	0.987	0.838	0.419	0.709	0.602	0.301	0.719
10	0.989	0.908	0.456	0.716	0.657	0.330	0.724
15	0.992	0.919	0.549	0.730	0.677	0.405	0.737
20	0.993	0.926	0.634	0.742	0.691	0.473	0.747

From Table 8.7 we can deduce that all the measurements of the efficiency increases with the number N of the points where the solution of the ODE system is numerical evaluated.

The complexity of the system function has a significant influence on the numerical efficiency (compare the values for $P = 2$ from both tables). For the test systems, the value of E_{num} is closed to 0.8.

The numerical efficiency is the principal value which has a great influence on the efficiency of the parallel algorithm. Hence, E_P depends on D, P and on the complexity of the system function.

Table 8.7: The efficiency in the integration of the Iserles's system (2) with the dimension $D = 2$ and $P = 2$ processors

$\lg(N)$	E_2^{par}	E_2	E_{num}
1	0.644	0.453	0.704
2	0.700	0.550	0.779
3	0.890	0.703	0.789
4	0.962	0.760	0.790

8.6.3 Conclusions

The first proposed parallel algorithm for distributed memory multiprocessors is designed to solve stiff systems of many ordinary differential equations. It has the advantage that the approximation error is the same like the one produced by the sequential algorithm.

The second proposed parallel algorithm is designed to solve linear stiff systems with variable coefficients and a perturbation depending on time. The approximation error is the same like in the sequential algorithm. The efficency of the parallel algorithm increases with the dimension and the complexity of the system function and decreases with the number of used processors.

Appendix A

Classification of the stiff systems

The numerical methods designed for solving stiff problems are not universally. Some numerical codes work better for a particular stiff system subclass, others for another subclasses.

The most popular classification of the stiff systems is due to ENRIGHT et al. [40]:

Class A: the linear systems with real matrix eigenvalues;

Class B: the linear systems with complex matrix eigenvalues;

Class C: the systems with nonlinear coupling between the smooth and stiff solution components;

Class D: the nonlinear systems with a real JACOBIAN's matrix eigenvalues;

Class E: the nonlinear systems with a complex JACOBIAN's matrix eigenevalues.

For proving the efficiency of a new method designed for stiff problems, the method must integrate with success all the stiff systems from STIFF-DETEST package [40].

In the next Tables we present the classical stiff systems. Note S the ratio between the maximum and minimum absolute eigenvalues evaluated at the starting point. The second notation correspond to that of STIFF-DETEST package.

Table A.1. Stiff linear systems from Class A

No.	**System**	T	$y(0)$	S
$S1$ $(A1)$	$\begin{cases} y_1' = -0.5y_1 \\ y_2' = -y_2 \\ y_3' = -100y_3 \\ y_4' = -90y_4 \end{cases}$	20	$\begin{matrix}1\\1\\1\\1\end{matrix}$	$2 \cdot 10^2$
$S2$	$\begin{cases} y_1' = 998y_1 + 1998y_2 \\ y_2' = -999y_1 - 1999y_2 \end{cases}$	100	$\begin{matrix}1\\0\end{matrix}$	10^3
$S3$	$\begin{cases} y_1' = -10^6 y_1 + 0.075y_2 \\ y_2' = 7500y_1 - 0.075y_2 \end{cases}$	10	$\begin{matrix}1\\-1\end{matrix}$	$7.4 \cdot 10^4$
$S4$ $(A2)$	$\begin{cases} y_1' = -1800y_1 + 900y_2 \\ y_i' = y_{i-1} - 2y_i + y_{i+1}, \quad i = 2(1)8 \\ y_9' = 1000y_8 - 2000y_9 + 10000 \end{cases}$	120	$\begin{matrix}0\\\vdots\\0\end{matrix}$	$2 \cdot 10^4$
$S5$ $(A3)$	$\begin{cases} y_1' = -10^4 y_1 + 100y_2 - 10y_3 + y_4 \\ y_2' = -10^3 y_2 + 10y_3 - 10y_4 \\ y_3' = -y_3 + 10y_4 \\ y_4' = -0.1y_4 \end{cases}$	20	$\begin{matrix}1\\1\\1\\1\end{matrix}$	10^5
$S6$ $(A4)$	$y_i' = -i^5 y_i, \ i = 1(1)10$	1	$\begin{matrix}1\\\vdots\\1\end{matrix}$	10^5
$S7$	$\begin{cases} y_1' = -k_1 y_1 + k_2 y_3 \\ y_2' = -k_4 y_2 + k_3 y_3 \\ y_3' = k_1 y_1 + k_4 y_2 - (k_2 + k_3) y_3 \\ k_1 = 8.43 \cdot 10^{-10}, \ k_2 = 2.9 \cdot 10^{11}, \\ \quad k_3 = 2.46 \cdot 10^{10}, \ k_4 = 8.76 \cdot 10^{-6} \end{cases}$	$8 \cdot 10^5$	$\begin{matrix}0\\1\\0\end{matrix}$	$3.146 \cdot 10^{11}$

Table A.2. Stiff linear systems from Class B

No.	**System**	T	$y(0)$	S
$S8$ $(B1)$	$\begin{cases} y_1' = -y_1 + y_2 \\ y_2' = -100y_1 - y_2 \\ y_3' = -100y_3 + y_4 \\ y_4' = -10000y_3 - 100y_4 \end{cases}$	20	$\begin{matrix}1\\0\\1\\0\end{matrix}$	10^2
$S9$ $(B2 - B5)$	$\begin{cases} y_1' = -10y_1 + \alpha y_2 \\ y_2' = -\alpha y_1 - 10y_2 \\ y_3' = -4y_3 \\ y_4' = -y_4 \\ y_5' = -0.5y_5 \\ y_6' = -0.1y_6 \end{cases}$ $\alpha = 3, 8, 25, 100$	20	$\begin{matrix}0\\1\\1\\1\\1\\1\end{matrix}$	10^2

Table A.3. Stiff perturbated linear systems from Classes A and B

No.	**System**	T	$y(0)$	S
$S10$	$\begin{cases} y' = -200(y - E) + E', \\ E(t) = 10 - (10 + t)e^{-t} \end{cases}$	100	10	$2 \cdot 10^2$
$S11$	$y' = ay + ade^{dt},\ a = 10^{-4},\ d = i \in \mathbf{C} - \mathbf{R}$	100	y_0	10^3
$S12$	$\begin{cases} y_1' = -6y_1 + 5y_2 + 2\sin t \\ y_2' = 94y_1 - 95y_2 \end{cases}$	100	$\begin{matrix} 0 \\ 0 \end{matrix}$	10^3
$S13$	$\begin{cases} y_1' = -4498y_1 - 5996y_2 + 0.006 - t \\ y_2' = 2248.5y_1 + 2997y_2 - 0.503 + 3t \end{cases}$	25	$\begin{matrix} \frac{25498}{1500} \\ -\frac{16499}{1500} \end{matrix}$	$1.5 \cdot 10^3$
$S14$	$\begin{cases} y_1' = -2000y_1 + 1000y_2 + 1 \\ y_2' = y_1 - y_2 \end{cases}$	100	$\begin{matrix} 0 \\ 0 \end{matrix}$	$4 \cdot 10^3$
$S15$	$y' = UBUy + UD,\ u_{ii} = -\frac{1}{3},\ u_{ij} = \frac{2}{3}, i \neq j$ $B = \begin{pmatrix} -100 & 1000 & 0 \\ -1000 & -100 & 0 \\ 0 & 0 & -0.1 \end{pmatrix}, D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	100	$\begin{matrix} -\frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{matrix}$	10^3
$S16$	$y' = UBU^T y + Uf(t),$ $u_{ii} = -\frac{1}{2},\ u_{ij} = \frac{1}{2}, i \neq j$ $B = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -100 & -900 \\ 0 & 0 & 900 & -100 \end{pmatrix},$ $f(t) = (t^2 + 2t, t^2 - 2t,$ $-800t + 1, -1000t - 1)^T$	1	$\begin{matrix} 1 \\ 0 \\ 0 \\ 1 \end{matrix}$	10^2

Table A.4. Stiff linear systems with variable coefficients from Class A

No.	**System**	T	$y(0)$	S
$S17$	$\begin{cases} y_1' = -(80 + \frac{1}{3}(1 + t)^{-1})y_1 - \\ \quad -(40 - \frac{2}{5}(1 + t)^{-1})y_2 \\ y_2' = -(40 - \frac{2}{5}(1 + t)^{-1})y_1 - \\ \quad -(20 + \frac{4}{5}(1 + t)^{-1})y_2 \end{cases}$	100	$\begin{matrix} 0 \\ 1 \end{matrix}$	10^2
$S18$ $(D1)$	$\begin{cases} y_1' = 0.2(y_2 - y_1) \\ y_2' = 10y_1 - (60 - 0.125t)y_2 + 0.125t \end{cases}$	400	$\begin{matrix} 0 \\ 0 \end{matrix}$	$3.53 \cdot 10^2$

Table A.5. Stiff nonlinear systems from Class C

No.	**System**	T	$y(0)$	S
$S19$ $(C1)$	$\begin{cases} y_1' = -y_1 + y_2^2 + y_3^2 + y_4^2 \\ y_2' = -10y_2 + 10(y_3^2 + y_4^2) \\ y_3' = -40y_3 + 40y_4^2 \\ y_4' = -100y_4 + 2 \end{cases}$	20	$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix}$	10^2
$S20$ $(C2)$	$\begin{cases} y_1' = -y_1 + 2 \\ y_2' = -10y_2 + 0.1y_1^2 \\ y_3' = -40y_3 + 4(y_1^2 + y_2^2) \\ y_4' = -100y_4 + y_1^2 + y_2^2 + y_3^2 \end{cases}$	20	$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \end{matrix}$	10^2

Table A.6. Stiff nonlinear systems from Class D

No.	**System**	T	$y(0)$	S
$S21$	$y_i' = -\beta_i y_i + y_i^2,\ (\beta_i)_{i=1(1)4} = \begin{pmatrix} 1000 \\ 800 \\ -10 \\ 0.001 \end{pmatrix}$	20	$\begin{matrix} -1 \\ -1 \\ -1 \\ -1 \end{matrix}$	10^6
$S22$ $(D2)$	$\begin{cases} y_1' = -0.04y_1 + 0.01y_2y_3 \\ y_2' = 400y_1 - 100y_2y_3 - 3000y_2^2 \\ y_3' = 30y_2^2 \end{cases}$	40	$\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$	10^5
$S23$	$\begin{cases} y_1' = -0.04y_1 + 10^4 y_2y_3 \\ y_2' = 0.04y_1 - 10^4 y_2y_3 - 3 \cdot 10^7 y_2^2 \\ y_3' = 3 \cdot 10^7 y_2^2 \end{cases}$	40	$\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$	10^5
$S24$ $(D3)$	$\begin{cases} y_1' = y_3 - 100y_1y_2 \\ y_2' = y_3 + 2y_4 - 100y_1y_2 - 2 \cdot 10^4 y_2^2 \\ y_3' = -y_3 + 100y_1y_2 \\ y_4' = -y_4 + 10^4 y_2^2 \end{cases}$	20	$\begin{matrix} 1 \\ 1 \\ 0 \\ 0 \end{matrix}$	$4 \cdot 10^2$
$S25$ $(D4)$	$\begin{cases} y_1' = -0.013y_1 - 1000y_1y_3 \\ y_2' = -2500y_2y_3 \\ y_3' = -0.013y_1 - 1000y_1y_3 - 2500y_2y_3 \end{cases}$	50	$\begin{matrix} 1 \\ 1 \\ 0 \end{matrix}$	$3 \cdot 10^5$
$S26$ $(D5)$	$\begin{cases} y_1' = 0.01 - (0.01 + y_1 + y_2) \cdot \\ \quad \cdot (y_1^2 + 1001y_1 + 1001) \\ y_2' = 0.01 - (0.01 + y_1 + y_2)(1 + y_2^2) \end{cases}$	100	$\begin{matrix} 0 \\ 0 \end{matrix}$	10^6
$S27$ $(D6)$	$\begin{cases} y_1' = -y_1 + 10^8 y_3(1 - y_1) \\ y_2' = -10y_2 + 3 \cdot 10^7 y_3(1 - y_2) \\ y_3' = -y_1' - y_2' \end{cases}$	1	$\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$	$3 \cdot 10^7$

Table A.7. Stiff nonlinear systems from Class E

No.	**System**	T	$y(0)$	S
$S28$ $(E2)$	$\begin{cases} y_1' = y_2 \\ y_2' = 5(1-y_1^2)y_2 - y_1 \end{cases}$	1	$\begin{matrix} 2 \\ 0 \end{matrix}$	$5.2 \cdot 10^6$
$S29$ $(E3)$	$\begin{cases} y_1' = -(55+y_3)y_1 + 65y_2 \\ y_2' = 0.0785(y_1 - y_2) \\ y_3' = 0.1y_1 \end{cases}$	500	$\begin{matrix} 1 \\ 1 \\ 0 \end{matrix}$	10^4
$S30$ $(E4)$	$y' = -By + U^T \begin{pmatrix} z_1^2/2 - z_2^2/2 \\ z_1 z_2 \\ z_3^2 \\ z_4^2 \end{pmatrix}$, $z = Uy$, $B = U \begin{pmatrix} \beta_1 & -\beta_2 & 0 & 0 \\ \beta_2 & \beta_1 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ 0 & 0 & 0 & \beta_4 \end{pmatrix} U$, $u_{ij} = \frac{1}{2}, i \neq j,\ u_{ii} = -\frac{1}{2}$, $\beta_1 = -\beta_2 = -10$, $\beta_3 = 1000,\ \beta_4 = 0.01$	1	$\begin{matrix} 0 \\ -2 \\ -1 \\ -1 \end{matrix}$	10^6
$S31$ $(E5)$	$\begin{cases} y_1' = -7.89 \cdot 10^{-10} y_1 - 1.1 \cdot 10^7 y_1 y_3 \\ y_2' = 7.89 \cdot 10^{-10} y_1 - 1.13 \cdot 10^9 y_2 y_3 \\ y_3' = 7.89 \cdot 10^{-10} y_1 - 1.1 \cdot 10^7 y_1 y_3 + \\ \quad +1.13 \cdot 10^3 y_4 - 1.13 \cdot 10^5 y_2 y_3 \\ y_4' = 1.1 \cdot 10^7 y_1 y_3 - 1.13 \cdot 10^3 y_4 \end{cases}$	1000	$\begin{matrix} 0.002 \\ 0 \\ 0 \\ 0 \end{matrix}$	$2.7 \cdot 10^7$
$S32$	$\begin{cases} y_1' = 10^4 y_1 y_3 + 10^4 y_2 y_4 \\ y_2' = -10^4 y_1 y_4 + 10^4 y_2 y_3 \\ y_3' = 1 - y_3 \\ y_4' = -y_4 - 0.5y_3 + 0.5 \end{cases}$	10	$\begin{matrix} 1 \\ 1 \\ -1 \\ 0 \end{matrix}$	10^4

Table A.8. Some stiff nonlinear systems which model real systems

No.	**System**	T	$y(0)$
$S33$	$\begin{cases} y_1' = -10^4 y_1 + y_2^4 - 2y_3^2 - y_4^2 - y_5 \\ y_2' = -\frac{1}{2} y_2 + y_1 - y_3^2 \\ y_3' = -0.01 y_2^2 \\ y_4' = -y_3 + y_1^3 - y_5^3 \\ y_5' = -y_1 - y_3 y_4 \end{cases}$	1	$\begin{matrix} 1 \\ 10 \\ 1 \\ 1 \\ 1 \end{matrix}$
$S34$	$\begin{cases} y_1' = -y_1 - y_1 y_2 + \varepsilon k y_2 \\ \varepsilon y_2' = y_1 - y_1 y_2 - \varepsilon k y_2 \end{cases}$ $\varepsilon = 98,\ k = 3$	100	$\begin{matrix} 1 \\ 1 \end{matrix}$
$S35$	$\begin{cases} y_1' = y_2 \\ y_2' = ((1 - y_1^2)\sqrt{\lvert 1 - y_1^2 \rvert} y_2 - y_1)/\varepsilon \end{cases}$ $\varepsilon = 10^{-6}$	11	$\begin{matrix} 2 \\ 0 \end{matrix}$
$S36$	$\begin{cases} y_1' = -(b + ai) y_1 + b y_2^i \\ y_2' = y_1 - a y_2 - y_2^i \end{cases},$ $a = 0.1, c = 1, b = 10^2, i = 4$	10	$\begin{matrix} c^i \\ c \end{matrix}$
$S37$	$\begin{cases} y_1' = 250[(-0.0048(y_3 - 660.2) - \\ \quad -0.032(y_5 - 273.9) - 1) y_1 + y_2] \\ y_2' = 0.1(y_1 - y_2) \\ y_3' = 93 y_1 - 0.26(y_3 - y_4) \\ y_4' = 0.87(y_3 - y_4) - 11(y_4 - y_5) \\ y_5' = 1.8(y_4 - y_5) - 13(y_5 - 270) \end{cases}$	1	$\begin{matrix} 1 \\ 1 \\ 660.2 \\ 302.2 \\ 273.9 \end{matrix}$
$S38$	$\begin{cases} y_1' = 100 y_1 / y_2 (y_3 - y_1) \\ y_2' = -100(y_3 - y_1) \\ y_3' = 1/y_4\ [0.9 - 1000(y_3 - y_5) - 100 y_3 (y_3 - y_1)], \\ y_4' = 100(y_3 - y_1) \\ y_5' = -100(y_5 - y_3) \end{cases}$ $a = 0.99026, 0.99, 0.9$	1	$\begin{matrix} 1 \\ 1 \\ 1 \\ -10 \\ a \end{matrix}$

Table A.9. Other stiff nonlinear systems which model real systems

No.	**System**	T	$y(0)$
$S39$	$\begin{cases} y_1' = 77.27[y_2 + y_1(1 - 8.375 \cdot 10^{-6} y_1 - y_2)] \\ y_2' = [y_3 - (1 + y_1)y_2]/77.27 \\ y_3' = 0.161(y_1 - y_3) \end{cases}$	300	3 1 2
$S40$	$\begin{cases} y_1' = -1.71y_1 + 0.43y_2 + 8.32y_3 + 7 \cdot 10^{-4} \\ y_2' = 1.71y_1 - 8.75y_2 \\ y_3' = -10.03y_3 + 0.43y_4 + 0.035y_5 \\ y_4' = 8.32y_2 + 1.71y_3 - 1.12y_4 \\ y_5' = -1.745y_5 + 0.43y_6 + 0.43y_7 \\ y_6' = -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7 \\ y_7' = 280y_6y_8 - 1.81y_7 \\ y_8' = -y_7' \end{cases}$	400	1 0 0 0 0 0 0 0.0057
$S41$	$\begin{cases} y_1' = -(2 + \varepsilon^{-1})y_1 + \varepsilon^{-1} y_2^2 \\ y_2' = y_1 - y_2(1 + y_2) \end{cases}, \ \varepsilon = 10^{-8}$	4	1 1
$S42$	$\begin{cases} y_1' = -k_1 y_1 - k_2 y_1 y_2 \\ y_2' = k_1 y_1 + k_3 y_3 - k_4 y_2 y_4 - 2k_5 y_2^2 \\ y_3' = k_2 y_1 y_2 - k_3 y_3 \\ y_4' = -k_4 y_2 y_4 \end{cases}$, $k_1 = 10^{-4}, \ k_2 = 2.9 \cdot 10^4$, $k_3 = 5 \cdot 10^3, \ k_4 = 10^4$, $k_5 = 6.7 \cdot 10^{10}$	20	0.6 0 0 0.4
$S43$	$\begin{cases} y_1' = 1.30(y_3 - y_1) + 2.13 \cdot 10^6 k(y_1) y_2 \\ y_2' = 1.80 \cdot 10^3 [y_4 - y_2(1 + k(y_1))] \\ y_3' = 1752 - 269y_3 + 267y_1 \\ y_4' = 0.1 + 320y_2 - 321y_4 \end{cases}$, $k(y_1) = 0.006 \cdot e^{20.7 - 15000/y_1}$	1	761 0 600 0.1

Appendix B

NEWTON's iterations error

B.1 Motivation

In the numerical treatement of the stiff equations we deal with some implicit formula of the following form:

$$y - h\beta f(y) = b$$

A convergent method, for which is not necessary that the system function f to safisfy the inequality $|h\partial f/\partial y| \leq c$, is the NEWTON's method. The classical convergence conditions are establish by the KANTOROVICH's theorem [94].

The problems associated with an implementation of the NEWTON's methods are the folowings:

1. the starting values;
2. a minimum effort in evaluating the derivative;
3. a method for solving the associated linear system;
4. the estimate of the error.

In this section we are treating only the problem 4.
Generally, we search the solution of the equation system

$$F(x) = 0,$$

where $F : D \subset \mathbf{R}^n \to \mathbf{R}^n$ is a given operator. The NEWTON's iterations are well defined by the process

$$x^{k+1} = x^k - F'(x^k)^{-1}F(x^k), \quad k = 0, 1, \ldots$$

where F' is the GATEAUX's derivative. Let $\|\cdot\|$ the Euclidian norm on $\mathbf{R}^n$. We use the same notation for the associated matrix norm.

Theorem B.1. *(NEWTON-KANTOROVICH's theorem [94]) Assume that $F : D \subset R^n \to R^n$ is F-differentiable on a convex set $D_0 \subset D$ and that*

$$\|F'(x) - F'(y)\| \leq \gamma \|x - y\|, \quad \forall x, y \in D_0.$$

Suppose that there exists an $x^0 \in D_0$ such that $\|F'(x^0)^{-1}\| \leq \beta$ and $\alpha = \beta\gamma\mu \leq \frac{1}{2}$, where $\mu \geq \|F'(x^0)^{-1}F(x^0)\|$. Set

$$t^* = (\beta\gamma)^{-1}[1 - (1 - 2\alpha)^{1/2}], \quad t^{**} = (\beta\gamma)^{-1}[1 + (1 - 2\alpha)^{1/2}]$$

and assume that $\bar{S}(x^0, t^*) \subset D_0$. Then the NEWTON's iterates are well-defined, remain in $\bar{S}(x^0, t^*)$, and converge to a solution x^* of $F(x) = 0$, which is unique in $S(x^0, t^{**}) \cap D_0$. Moreover, the error estimate

$$\|x^* - x^k\| \leq (\beta\gamma 2^k)^{-1}(2\alpha)^{2^k}, \quad k = 0, 1, \ldots$$

holds.

Note that the theorem assumes that the derivative is LIPSCHITZ continous.

Counterexample. Let the CAUCHY's problem

$$\begin{cases} y_1' = -\sqrt{y_2^3}, \\ y_2' = -\sqrt{y_1^3}, \\ y_1(1) = y_2(1) = 4, \end{cases}$$

with at most one solution $y_1(t) = y_2(t) = 4/t^2$, $t > 1$, which is numerical integrated with the EULER's implicit rule. Assume that, at one step, $y_{n-1} = (4/h^2, 8/h^2)^T$. Note $y_n = (h^2 x_1, h^2 x_2)$. Then the system is

$$\begin{cases} x_1 + \sqrt{x_2^3} = 4, \\ \sqrt{x_1^3} + x_2 = 8, \end{cases}$$

with the exact solution $x_1 = 4, x_2 = 0$. Therefore $y_n = (4h^2, 0)$. The system function on the unknown x does not satisfies the LIPSCHITZ's condition on the region $[0, \infty) \times [0, \infty)$, but satisfies the HÖLDER's inequality

$$\|F'(x) - F'(y)\| \leq \gamma \|x - y\|^p, \quad \forall x, y \in D_0, \quad 0 \leq p \leq 1$$

with $\gamma = 3/2$ and $p = 1/2$ [113].

In the case of a HÖLDER continous derivative, the NEWTON attraction theorem [94] tell us only that the iterations converge with an superlinear spead.

An other theorem for estimating the error is the following.

Theorem B.2. *(NEWTON-MYSOVSKII theorem [94]) Suppose that $F : D \subset R^n \to R^n$ is F-differentiable on a convex set $D_0 \subset D$ and that for each $x \in D_0$, $F'(x)$ is nonsingular and satisfies the LIPSCHITZ condition and $\|F'(x)^{-1}\| \leq \beta, \forall x \in D_0$. If $x^0 \in D_0$ is such that $\|F'(x^0)^{-1}F(x^0)\| \leq \mu$ and $\alpha = \frac{1}{2}\beta\gamma\mu < 1$, as well as $\bar{S}(x^0, r_0) \subset D_0$, where*

$$r_0 = \mu \sum_{j=0}^{\infty} \alpha^{2^j - 1}$$

then the NEWTON*'s iterates remain in* $\bar{S}(x^0, r_0)$ *and converge to a solution* x^* *of* $F(x) = 0$*. Moreover, the following estimate holds*

$$\|x^* - x^k\| \leq \epsilon_k \|x^k - x^{k-1}\|^2, \quad k = 1, 2, \ldots$$

where

$$\epsilon_k = \frac{\alpha}{\mu} \sum_{j=0}^{\infty} (\alpha^{2^k})^{2^j - 1} \leq \alpha [\mu(1 - \alpha^{2^k})]^{-1}, \quad k = 1, 2, \ldots$$

B.2 Generalization of the classical theorems

We propose the following generalization of the NEWTON-MYSOVSKII theorem:

Theorem B.3. *[101] Suppose that* $F : D \subset R^n \to R^n$ *is F-differentiable on a convex set* $D_0 \subset D$ *and that for each* $x \in D_0$*,* $F'(x)$ *is nonsingular and satisfies the* HÖLDER *condition with* $p \leq 1$ *and* $\|F'(x)^{-1}\| \leq \beta, \forall x \in D_0$*. If* $x^0 \in D_0$ *is such that* $\|F'(x^0)^{-1} F(x^0)\| \leq \mu$ *and* $\alpha = \frac{1}{p+1} \beta \gamma \mu^p < 1$*, as well as* $\bar{S}(x^0, r_0) \subset D_0$*, where*

$$r_0 = \mu \sum_{j=0}^{\infty} \alpha^{\frac{(p+1)^j - 1}{p}}$$

then the NEWTON*'s iterates remain in* $\bar{S}(x^0, r_0)$ *and converge to a solution* x^* *of* $F(x) = 0$*. Moreover, the following estimate holds*

$$\|x^* - x^k\| \leq \epsilon_k \|x^k - x^{k-1}\|^{p+1}, \quad k = 1, 2, \ldots$$

where

$$\epsilon_k = \frac{\alpha}{\mu^p} \sum_{j=0}^{\infty} (\alpha^{\frac{(p+1)^k}{p}})^{\frac{(p+1)^j - 1}{p}} \leq \alpha \mu^{-p} \left(1 - \alpha^{\frac{(p+1)^k}{p}}\right)^{-1}, \quad k = 1, 2, \ldots$$

For the KANTOROVICH's theorem, we have find a partial generalization.

Theorem B.4. *[97] Assume that* $F : D \subset R^n \to R^n$ *is F-differentiable on a convex set* $D_0 \subset D$ *and the* HÖLDER*'s inequality holds for* $p \leq 1$*. Suppose that there exists an* $x^0 \in D_0$ *such that* $\|F'(x^0)^{-1}\| \leq \beta$ *and* $\alpha = \beta \gamma \mu^p \leq \frac{p}{p+1}$*, where* $\mu \geq \|F'(x^0)^{-1} F(x^0)\|$ *and* $\bar{S}(x^0, (\beta\gamma)^{-1/p}) \subset D_0$*. Then the* NEWTON*'s iterates are well-defined, remain in* $\bar{S}(x^0, (\beta\gamma)^{-1/p})$*, and converge to a solution* x^* *of* $F(x) = 0$*. Moreover, the error estimate*

$$\|x^* - x^k\| \leq \frac{1}{p} \left(\frac{p}{\beta\gamma}\right)^{\frac{1}{p}} (p+1)^{-k} \left[\frac{(p+1)^p \alpha}{p}\right]^{\frac{(p+1)^k}{p}}, \quad k = 0, 1, \ldots$$

holds.

In the case $p = 1$ we get the NEWTON-KANTOROVICH's error estimate.

Example. We apply the last theorem to our above mention counterexample. The error estimate is

$$\|x^* - x^k\| \leq \frac{2}{9}\|F'(x^0)^{-1}\|^{-2}\left(\frac{2}{3}\right)^{(\frac{3}{2})^k+k}$$

for any value x^0 chosen so that

$$\|F'(x^0)^{-1}\|_2^3\|F(x^0)\|_2 \leq \frac{4}{81}$$

holds [105].

Appendix C

Computing the stability angle

We propose an algorithm for finding the minimum value of a real function.

C.1 Motivation

If the curve, which describes the stability domain boundary of a stiffly-stable formula, can be expressed in the parametric form

$$\theta \in [0, 2\pi) \longrightarrow z(\theta) \in \mathbf{C},$$

then the stability angle is the value α for which

$$g(\alpha) = \min_{\theta \in [0,\pi]} g(\theta) \ ,$$

where

$$g(\theta) = \begin{cases} \frac{\pi}{2}, \ \theta = 0, \ \mathrm{Re}\, z(\theta) = 0, \\ \pi, \ \theta = \pi, \ \mathrm{Re}\, z(\theta) = 0, \\ \mathrm{arctg} \left| \dfrac{\mathrm{Im}\, z(\theta)}{\mathrm{Re}\, z(\theta)} \right|, \ \mathrm{Re}\, z(\theta) < 0, \\ \dfrac{\pi}{2} + \mathrm{arctg} \left| \dfrac{\mathrm{Im}\, z(\theta)}{\mathrm{Re}\, z(\theta)} \right|, \ \mathrm{Re}\, z(\theta) > 0 \end{cases} .$$

Therefore, the problem of computing the stability angle is a problem of onedimensional minimization.

Remark. The boundary curve of a linear multistep formula generated by the couple (ρ, σ) is

$$\theta \in [0, 2\pi) \longrightarrow z(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta}).}$$

If we deal with a multiderivative linear formula, the characterisc equation has a polynomial expression and there is a number of curves, equal to the derivative maximum degree, which can be expressed as some $z(\theta)$ curves.

C.2 Onedimensional minimization

Let the problem to minimize the real function $g : [a_0, b_0] \to R$, i.e. finding

$$\min_{x \in [a_0, b_0]} g(x).$$

When the g expression is very complicated, it is preferable to use a method which does not request the derivative evaluation. Such methods are the *gold section* method or the FIBONACCI *search method.*

We propose an improvement of those methods.

C.3 λ-algorithm

The two above mentionated methods can be expressed in a form named λ-algorithm [94]. For each $\lambda \in [\frac{1}{2}, 1]$, the λ-algorithm follows the steps:

Step 1. By a certain method, we determine an interval $[a_0, b_0]$, which includes the solution x^*. We give the error tolerance ϵ. Set

$$\Delta_0 = b_0 - a_0, \Delta_1 = \lambda\Delta_0, \Delta_2 = \Delta_0 - \Delta_1, y_0 = a_0 + \Delta_0, z_0 = b_0 - \Delta_2, k = 0$$

We compute $g(y_0)$ and $g(z_0)$.

Step 2. If $\Delta_k \leq \epsilon$, then we put $x^* = arg\,min\,\{g(y_k), q(z_k)\}$ and we stop the algorithm, else $k \leftarrow k + 1$.

Step 3. We put $\Delta_{k+2} = \Delta_k - \Delta_{k+1}$ and, if $g(y_{k-1}) \leq g(z_{k-1})$, then

$$a_k = a_{k-1}, b_k = z_{k-1}, z_k = y_{k-1}, y_k = a_k + \Delta_{k+2},$$

else

$$a_k = y_{k-1}, b_k = b_{k-1}, y_k = z_{k-1}, z_k = b_k - \Delta_{k+2}.$$

Go to step 2.

We can make the following remarks concerning this class of algorithms:

1. $\Delta_k = b_k - a_k = (-1)^{k-1}(F_k\lambda - F_{k-1})\Delta_0,\ k \geq 2$, where F_k is the FIBONACCI sequence starting with $F_1 = F_2 = 1$ and $F_{k+2} = F_{k+1} + F_k$;

2. the conditions $[a_k, b_k] \subset [a_{k-1}, b_{k-1}],\ k = 1, \ldots, n$ are equivalent to certain conditions for λ. Note $\lambda_{k+1} = F_k/F_{k+1}$. Then

$$\begin{aligned} &\lambda_n < \lambda < \lambda_n, \text{ if } n \text{ is odd} \\ &\lambda_{n+1} < \lambda < \lambda_n, \text{ otherwise} \end{aligned} \qquad (C.1)$$

3. we get the gold section method when $\lambda = \lim_{n\to\infty} \lambda_n = (\sqrt{5}-1)/2$ and then $\Delta_k = \lambda^k \Delta_0$. The stepnumber n is given by

$$\lambda^n \Delta_0 < \epsilon$$

The method is optimal from the point of view of the calculus effort;

4. the FIBONACCI search method is obtained when $\lambda = \lambda_n$ and then $\Delta_k = \Delta_0 F_{n-k}/F_n$. The step number is given by

$$\Delta_0/F_n < \epsilon \qquad (C.2)$$

Moreover, the method is optimal from the point of view of the error in the above mentioned class, because, for any λ satisfying (C.1), the following inequalities hold:

$$0 < \Delta_{n-2}(\lambda_n) < \Delta_{n-2}(\lambda)$$

5. the algorithms have linear convergence;

6. they are not stable with respect to the errors on estimating the permissible values of λ. If $\lambda = \lambda' + \delta$, then

$$\Delta_k(\lambda) = \Delta_k(\lambda') + (-1)^{k-1}\delta F_k \Delta_0$$

It is thus necessary to stipulate besides the error level the control of this error, i.e., λ must be computed with an approximation less then $\epsilon/(F_k\Delta_0)$;

7. for $n \geq 25$ the values of λ for the gold section and for the FIBONACCI search method are the same in the first 10 decimal digits;

8. to counteract the shortcomings of the last two remarks, we can adopt a modified variant, using the FIBONACCI search method in several cycles. One cycle consists of running trough $n-2$ steps with the standard method. Then we restart the process with $a_0 = a_{n-2},\ b_0 = b_{n-2}$. After p cycles we get

$$\Delta_{p(n-2)}(\lambda) < \Delta_0/F_n^p$$

9. both methods assume a certain symmetry, which ensures the independence of the sequence of approximation intervals from the initial interval;

10. the convergence conditions implies some constraints concerning λ at each step: as upper bound at a certain step, and as lower bound at the next.

C.4 (α, β)-algorithm

We assume that we have two parameters. We expect that the conditions for the decrease of the approximation interval requires an upper bound for one of those parameters, and a lower bound for the other, and the calculus effort remain rather identical.

The question if we can obtain a more efficient method, in the sense to reduce the number of objective function evaluation. The answer is affirmative, as will be proved in the following. We can see in the numerical examples that the number of estimates of the objective function g is lowest than the FIBONACCI search method, but in addition a recursive calculus of an other Δ_k is request at each step.

We consider the following (α, β)-algorithm, where $\alpha, \beta \in [\frac{1}{2}, 1]$:

Step 1. By a certain method, we determine an interval $[a_0, b_0]$ which includes the solution x^*. We give the error tolerance ϵ. Set

$$\Delta_0 = b_0 - a_0,\ \Delta_1(\alpha) = \alpha\Delta_0,\ \Delta_1(\beta) = \beta\Delta_0,$$

$$\Delta_2(\alpha) = \Delta_0 - \Delta_1(\alpha),\ \Delta_2(\beta) = \Delta_0 - \Delta_1(\beta),$$

$$y_0 = a_0 + \Delta_0(\alpha),\ z_0 = b_0 - \Delta_2(\alpha),\ k = 0$$

We compute $g(y_0)$ and $g(z_0)$.

Step 2. If $b_k - a_k \leq \epsilon$, then we put $x^* = arg\min\{g(y_k), q(z_k)\}$ and we stop the algorithm, else $k \leftarrow k + 1$.

Step 3. We put $\Delta_{k+2}(\alpha) = \Delta_k(\alpha) - \Delta_{k+1}(\alpha)$, $\Delta_{k+2}(\beta) = \Delta_k(\beta) - \Delta_{k+1}(\beta)$, and, if $g(y_{k-1}) \leq g(z_{k-1})$, then

$$a_k = a_{k-1},\ b_k = z_{k-1},\ z_k = y_{k-1},\ \begin{cases} y_k = a_k + \Delta_{k+2}(\alpha), & \text{if } k \text{ is odd} \\ y_k = a_k + \Delta_{k+2}(\beta), & \text{if } k \text{ is even} \end{cases},$$

else

$$a_k = y_{k-1},\ b_k = b_{k-1},\ y_k = z_{k-1},\ \begin{cases} z_k = b_k - \Delta_{k+2}(\alpha), & \text{if } k \text{ is odd} \\ z_k = b_k - \Delta_{k+2}(\beta), & \text{if } k \text{ is even} \end{cases},$$

Go to step 2.

C.5 Properties of the (α, β)-algorithm

Concerning the proposed algorithm we make the following statements.

Lemma C.1. *If $\alpha \leq \beta$, then, for any $k \geq 0$, the inequalities*

$$\Delta_{2k+1}(\alpha) \leq \Delta_{2k+1}(\beta), \quad \Delta_{2k}(\beta) \leq \Delta_{2k}(\alpha)$$

hold.

To prove this statement we use the first remark from the previous section.

Lemma C.2. *For any $k \geq 2$, the equality*

$$\sum_{i=1}^{k-1} (-1)^i \left(\Delta_i(\beta) - \Delta_i(\alpha)\right) = -\left(F_{k+1} - 1\right)(\beta - \alpha)\Delta_0$$

holds.

To prove this statement we use the recurrence relation for the FIBONACCI's sequence.

The two lemmas lie to the following more general statement.

Proposition C.1. *[104] If $\alpha \leq \beta$, then for any $k \geq 0$, the inequalities*

$$\min\left\{\Delta_k(\alpha), \Delta_k(\beta)\right\} - \left(F_{k+1} - 1\right)(\beta - \alpha)\Delta_0 \;\leq\; b_k - a_k \;\leq\; \max\left\{\Delta_k(\alpha), \Delta_k(\beta)\right\}$$

$$\min\left\{\Delta_{k+2}(\alpha), \Delta_{k+2}(\beta)\right\} \;\leq\; y_k - a_k, b_k - z_k \;\leq\; \max\left\{\Delta_{k+2}(\alpha), \Delta_{k+2}(\beta)\right\}$$

hold. Moreover, for each strictly unimodal function q, there is a pair (a_0, b_0) for which the left equality from the first relationship holds, and another for which the right equality holds.

The question is whether there are in this new class some algorithms more efficient than the FIBONACCI search method. First of all, that implies the stop of the new algorithm no later than the $(n-2)^{th}$ step, where n satisfies the inequality (C.2).

Proposition C.2. *[104] If n verifies the condition (C.2), and*

$$\alpha = \lambda_n \;,\; \alpha \leq \beta \leq \lambda_n + \frac{1}{F_n(F_n - 1)},\; \textit{for } n \textit{ even}$$
$$\beta = \lambda_n \;,\; \lambda_n - \frac{1}{F_n(F_n - 1)} \leq \alpha \leq \beta \;,\; \textit{for } n \textit{ odd}$$

then

$$b_k - a_k \geq 0, \quad k = 0, \ldots, n-1$$
$$0 \leq b_{n-1} - a_{n-1} \;,\; b_{n-2} - a_{n-2} \leq \Delta_{n-2}(\lambda_n)$$

hold.

Drawing the conclusions, we can make the followings statements:

1. the sequence of the approximation interval lenghts consists on only positive numbers and it is superior bounded by a sequence of numbers which decrease to zero. Consequently, for $n \to \infty$, the sequence of approximations, given by this method, is leading to the correct solution;

2. we have obtained a class of methods depending on a parameter, which gives the approximate solution bellow the error tolerance level no latter than the step, where the FIBONACCI search method stops;

3. for a given function g the approximation interval lenghts depend on the position of the initial values a_0, b_0 and the ideal solution of the problem;

4. note that, for $\alpha = \beta = \lambda_n$, we obtain the FIBONACCI search method.

We have a whole class. The problem is to choose the *optimal method*. Probably, we reach this purpose when $b_{n-2} - a_{n-2} = 0$. This equality holds when

$$\begin{cases} \alpha = \lambda_n,\ \beta = \lambda_n + \dfrac{1}{F_n(F_n - 1)}, & \text{if } n \text{ is even} \\ \beta = \lambda_n,\ \alpha = \lambda_n - \dfrac{1}{F_n(F_n - 1)}, & \text{if } n \text{ is odd} \end{cases} \tag{C.3}$$

Moreover, for these values $b_{n-1} - a_{n-1} = 0$ holds.

The following result proves the efficiency of the proposed method.

Theorem C.1. *[96] For a given strictly unimodal function g and an (α, β)-algorithm with the parameters satisfying (C.3), and n satisfying (C.2), there is a set of (ϵ, a_0, b_0) for which the algorithm is stopped at the $(n-3)^{th}$ step.*

Application. Let the problem to find the stability angle of the GEAR's backward differentiation formulae. We have use the p-cyclic versions with $n = 10$ for both λ and (α, β)-algorithms. The initial interval is $[0, \pi]$, and the error tolerance, $\varepsilon = 10^{-8}$. The results are presented in Table

Table C.1: Approximative values of the stability angles.for GEAR's formulae

Formula stepnumber	λ-algorithm stepnumber	Angle	(α, β)-algorithm stepnumber	Angle
3	48	86.0324	46	86.0324
4	48	73.3517	47	73.3517
5	48	51.8398	47	51.8398
6	48	17.8398	47	17.8398

Bibliography

[1] Aiken,C.(editor), *Stiff Computation*, Oxford University Press, 1985.

[2] Albrecht,P., Numerical Treatement of ODEs: The Theory of A-Methods, *Numerische Mathematik* 47 (1985) 59-87.

[3] Alfeld,P., Lambert,J.D., Correction in the Dominant Space: A Numerical Techniqe for a Certain Class of Stiff Initial Value Problems, *Mathematics of Computation* 31 (1977) 922-938.

[4] Bader,G., Deuflhard,P., A Semi-Implicit Mid-Point Rule for Stiff Systems of ODE, *Numerische Mathematik* 41 (1983) 373-398.

[5] Beaudet,P.R., Multi-off-grid Methods in Multi-step-integration of Ordinary Differential Equation, in [6].

[6] Bettis,D.G.,(editor) Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations, 19-20 October 1972, The University of Texas at Austin, Springer, *LNM 362*, 1974.

[7] Bickart,T.A., Jury,E.I., Arithmetic Test for A-stability, A(α)-stability and Stiff-stability, *BIT* 18 (1978) 9-21.

[8] Bickart,T.A., Picel,Z., High Order Stiffly Stable Composite Methods for Numerical Integration of Stiff Differential Equations, *BIT* 13 (1973) 271-286.

[9] Bickart,T.A., Rubin,W., Composite Multistep Methods and Stiff Stability, in [124].

[10] Bjorck,A., Some Methods for Separating Stiff Components in Initial Value Problems, in [50]

[11] Bjurel,G., Modified Linear Multistep Methods for a Class of Stiff Ordinary Differential Equations, *BIT* 12 (1972) 142-160.

[12] Brown,R.L., Some Characteristics of Implicit Multistep Multi-Derivative Integration Formulas, *SIAM Journal of Numerical Analysis* 14 (1977) 982-993.

[13] Bulirsch,R.,Grigorieff,R.D., (editors) *Numerical Treatement of Differential Equations*, Proceedings of a Conference Held at Oberwolfach, july 4-10, 1976, Springer, *LNM* 631, 1978.

[14] BURRAGE,K., CHIPMAN,F,H., The Stability Properties of Singly-Implicit General Linear Methods, *IMA Journal of Numerical Analysis* 5 (1985) 287-295.

[15] BUTCHER,J.C., Order Conditions for a General Class of Numerical Methods for Ordinary Differential Equations, in [84].

[16] BUTCHER,J.C., *The Numerical Analysis of Ordinary Differential Equations. Runge-Kutta and General Linear Methods*, John Willey and Sons Inc., Chichester, 1987.

[17] CARROLL,J., A Composite Integration Scheme for the Numerical Solution of Systems of Ordinary Differential Equations, *Journal of Computational and Applied Mathematics* 25 (1989) 1-13

[18] CASH,J.R., A Note on the Exponential Fitting of Blended, Extended Linear Multistep Methods, *BIT* 21 (1981) 450-454.

[19] CASH,J.R., On Improving the Absolute Stability of Local Extrapolation, *Numerische Mathematik* 40 (1982) 329-337.

[20] CASH,J.R., On the Design of High Order Exponentially Fitted Formulae for the Numerical Integration of Stiff Systems, *Numerische Mathematik* 36 (1981) 253-266.

[21] CASH,J.R., On the Exponential Fitting of Composite, Multiderivative Linear Multistep Methods, *SIAM Journal of Numerical Analysis* 18 (1981) 808-821.

[22] CASH,J.R., On the Integration of Stiff Systems of ODEs Using Extended Backward Differentiation Formulae, *Numerische Mathematik* 34 (1980) 235-246.

[23] *Cash,J.R.*, Second Derivative Extended Backward Differetiation Formulas for the Numerical Integration of Stiff Systems, *SIAM Journal on Numerical Analysis* 18 (1981) 21-36.

[24] CASH,J.R., Split Linear Multistep Methods for the Numerical Integration of Stiff Differential Systems, *Numerische Mathematik* 42 (1983) 299-310.

[25] CASH,J.R., *Stable Recursions with Applications to the Numerical Solutions of Stiff Systems*, Academic Press, 1979.

[26] CHAKRAVARTI,P.C., KANEL,M.S., Stiffly Stable Second Derivative Multistep Methods With Higher Order and Improved Stability Region, *BIT* 23 (1983) 75-83.

[27] CHARTRES,B., STEPLEMAN,R., A General Theory of Convergence for Numerical Methods, *SIAM Journal on Numerical Analysis* 9 (1972) 450-467.

[28] *Chu,M.T.*, An Automatic Multistep Method for Solving Stiff Initial Value Problems, *Journal of Computational and Applied Mathematics* 9 (1983) 229-238.

[29] CHU,M.T., HAMILTON,H., Parallel Solution of ODE's by Multi-Block Methods, *SIAM J.Sci. Statistic Comput.*, 27 (1978) 413-420.

[30] COOKE,C.H., On Stiffly Stable Implicit Linear Multistep Methods, *SIAM Journal on Numerical Analysis* 9 (1972) 45-57.

[31] COOPER,C.J., The Order of Convergence of General Linear Methods for Ordinary Differential Equations, *SIAM Journal of Numerical Analysis* 15 (1978) 643-661.

[32] CRYER,C.W., A New Class of Highly Stable Methods- A_0-stable Methods, *BIT* 13 (1973) 153-159.

[33] CRYER,C.W., On the Instability of High Order Backward-Difference Multistep Methods, *BIT* 12 (1972) 17-25.

[34] DAHLQUIST,G., The Sets of Smooth Solutions of Differential Difference Equations, in [124]

[35] DEUFLHARD,P. Recent progress in extrapolation methods for ordinary differential equations, *SIAM Review* 27 (1985) 505-535.

[36] DILL,G., GEAR,C.W., A Graphical Search for Stiffly Stable Methods for Ordinary Differential Equations, *Journal of Association for Computing Machinery* 18 (1971) 75-79.

[37] ENGLAND, R., Some Hybrid Implicit Stiffly Stable Methods for Ordinarry Differential Equations, in [56].

[38] ENRIGHT,W., Second Derivative Multistep Methods for Stiff Ordinary Differential Equations, *SIAM J. Numerical Analysis* 11 (1974) 321-331.

[39] ENRIGHT,W.H., Optimal Second Derivative Methods for Stiff Systems, in [124].

[40] ENRIGHT,W.H., HULL,T.E., LINDBERG,B., Comparing Numerical Methods for Stiff System of ODE's, *BIT* 15 (1975) 10-48.

[41] EVANS,D.J., MEGSON,G.M., Construction of Extrapolation Tables by Systolic Arrays for solving Ordinary Differential Equations, *Parallel Computing* 4 (1987) 33-48.

[42] EVANS,D.J., SANUGI,B.B., A Parallel Runge-Kutta Integration Method, *Parallel Computing* 11 (1989) 245-251.

[43] FATUNLA,S.O., *Numerical Methods for Initial Value Problems in Ordinary Differential Equations,* Academic Press, 1988.

[44] GALLIGAVI,I., RUGGIERO,V., Solving Large Systems of Linear Ordinary Differential Equations on a Vector Computer, *Parallel Computing* 9 (1989) 359-365.

[45] GEAR,C.W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey, 1971.

[46] GEAR,C.W., TU,K.W., The Effect of Variable Mesch Size on The Stability of Multistep Methods, *SIAM Journal of Numerical Analysis* 11 (1974) 1025-1043.

[47] GEAR,C.W., WATANABE,D.S., Stability and Convergence of Variable Order Multistep Methods, *SIAM Journal of Numerical Analysis* 11 (1974) 1044-1062.

[48] GENIN,Y., An Algebraic Approach to A-stable Linear Multistep-Multiderivative Integration Formulas, *BIT* 14 (1974) 382-406.

[49] GHOSHAL,S.K., GUPTA,M., RAJARAMAN,V., A Parallel Multistep Predictor-Corector Algorithm for Solving Ordinary Differential Equations, *Journal of Parallel and Distributed Computing* 6 (1989) 636-648.

[50] GRIFFITHS,D.F.,(editor) *Numerical Analysis*, Proceedings on the 10th Biennal Conference held at Dundee, Scotland, June 28-July 1, 1983, Springer, 1984.

[51] GRIGORIEFF,R.D., SCHROLL,J., Uber A(α)-stabile Verfahren hoher Konsistenzordung, *Computing* 20 (1978) 343-350.

[52] HAIRER,E., NORSETT,P.P., WANNER,G., *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer- Verlag, 1987.

[53] HAIRER,E., WANNER,G., *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer- Verlag, 1991.

[54] HAIRER,E., WANNER,G., Multistep-Multistage-Multiderivative Methods for Ordinary Differential Equations, *Computing* 11 (1973) 287-303.

[55] HAIRER,E., WANNER,G., On the Instability of the BDF Formulas, *SIAM Journal of Numerical Analysis* 20 (1983) 1206-1210.

[56] HENNART,J.H.,(editor) *Numerical Analysis*, Proceedings of the Third IIMAS Workshop Held at Cocoyoc, Mexico, Jan. 1981, *LNM* 909, Springer- Verlag, 1982.

[57] HOFER,E., A Partially Implicit Method for Large Stiff Systems of ODEs with Only Few Equations Introducing Small Time-Constants, *SIAM Journal of Numerical Analysis* 13 (1976) 534-567.

[58] HUTCHINSON,D., KHALAF,B.M.S., Parallel Algorithms for Solving Initial Value Problems: Front Broadering and Embedded Parallelism, *Parallel Computing* 17 (1991) 957-968.

[59] ISERLES,A., A Stability and Dominating Pairs, *Mathematics of Computation* 32 (1978) 19-33.

[60] ISERLES,A., NORSETT, S.P., On the Theory of Parallel Runge-Kutta Methods, *IMA Journal of Numerical Analysis* 10 (1990) 463-488.

[61] JACKSON,L.W., The A-stability of a Family of Fourth Order Methods, *BIT* 16 (1976) 383-387.

[62] JELTSCH,R., A_0-Stability and Stiff Stability of Brown's Multistep Muliderivative Methods, *Numerische Mathematik* 32 (1979) 167-181.

[63] JELTSCH,R., A Necessary Condition for A-Stability of Multistep Multiderivative Methods, *Mathematics of Computation* 30 (1976) 739-746.

[64] JELTSCH,R., Corrigendum: Stiff Stability of Multistep Multiderivative Methods, *SIAM Journal of Numerical Analysis* 16 (1979) 339-347.

[65] JELTSCH,R., On the Stability Region of Multistep Multiderivative Methods, in [13]

[66] JELTSCH,R., Stability on the Imaginary Axis and A-stability of the Linear Multistep Methods, *BIT* 18 (1978) 170-174.

[67] JELTSCH,R., Stiff Stability and Its Relation to A_0 and $A(\alpha)$-stability, *SIAM Journal of Numerical Analysis* 13 (1976) 8-17.

[68] JELTSCH,R., NEVANLINNA,O., Accuracy and Stability of Multistage Multistep Formulas, *Numerische Mathematik* 48 (1986) 33-83.

[69] KARAKASHIAN,O.A., RUST,W., On the Parallel Implementation of Implicit Runge-Kutta Methods, *SIAM Rev.* (1989) 1023-1028.

[70] KATZ,I.N., FRANKLIN,M.A., SEN,A., Optimally Stable Predictors for Adams-Moulton Correctors, *Comp. & Maths. with Appls.*, 3 (1977) 217-233.

[71] KNIRSCH, R., A parallel Implicit Runge-Kutta Method, in BADER,G., RANNACHER,R., WITTUM,G., Parallel Solution Methods for Differential Equations, *Preprint 696, Universität Heidelberg, Stochastiche Mathematische Modelle* (1992)

[72] LAMBERT,J.D., Linear Multistep Methods with Mildly Varying Coefficients, *Mathematics of Computation* 24 (1970) 81-93.

[73] LAMBERT,J.D., Nonlinear Methods for Stiff Systems of Ordinary Differential Equations, in Conference on the Numerical Solution of Differential Equations, Dundee, 1973, LNM.

[74] LAMBERT,J.D., Predictor-Corrector Algorithms with Identical Regions of Stability, *SIAM Journal of Numerical Analysis* 8 (1971) 337-344.

[75] LAMBERT,J.D., SIGURDSSON,S.T., Multistep Methods with Variable Matrix Coefficients, *SIAM Journal of Numerical Analysis* 9 (1972) 715-733.

[76] LAUTSCH,M., An Implicit Off-Step Point Method for the Integration of Stiff Differential Equations, *Computing* 31 (1983) 177-183.

[77] LEE,D., PREISER,S., A Class of Nonlinear Multistep Numerical A-stable Methods for Solving Stiff Differential Equations, *Computers and Mathematics with Applications* 4 (1978) 45-65.

[78] LI,S., RUAN,B., Nonlinear Stability of Multistep Multiderivative Methods, *Mathematics of Computation* 55 (1990) 581-589.

[79] LINDBERG,B., A Stiff System Package based on the Implicit Midpoint with Smoothing and Extrapolation, in [124].

[80] LINIGER,W.,WILLOUGHBY,R., Efficient Methods for solving Stiff Ordinary Differential Equation Systems, *SIAM Journal of Numerical Analysis* 7 (1970) 47-66.

[81] LINIGER,W., The A-contractive Second-Order Multistep Formulas with Variable Steps, *SIAM Journal of Numerical Analysis* 20 (1983) 1231-1238.

[82] MAJDA,G., Filtering Tehniques for Systems of Stiff Ordinary Differential Equations, *SIAM Journal of Numerical Analysis* 21 (1984) 535-566.

[83] MAKULA,M., NEVANLINNA,O., SIPILA,A.N., Exponentially Fitted Multistep Methods by Generalized Hermite-Birkchoff Interpolation, *BIT* 14 (1974) 437-451.

[84] MILLER,J.J.H.,(editor) *Topics in numerical analysis*, Proceedings of the Royal Irish Academy, Conference on Numerical Analysis, 1972, Academic Press, 1973.

[85] MIRANKER,W.L., *The Computational Theory of Stiff Differential Equations*, Université de Paris, Publication Mathématique D'Orsay, 1969.

[86] MIRANKER,W.L., *Numerical Methods for Stiff Equations and Singular Perturbation Problems*, D.Reidel Publishing Company, Mathematics and Its Applications 5, 1981.

[87] MIRANKER,W.L., LINIGER,W., Parallel Methods for the Numerical Integration of Ordinary Differential Equations, *Mathematics of Computation* 21 (1967) 303-320.

[88] MIRANKER,W.L., A Survey of Parallelism in Numerical Analysis, *SIAM Rev.* 13 (1971) 524-547.

[89] MORRIS,J.L.I.,(editor) Conference on the Numerical Solution of Differential Equations, Held in Dundee, Scotland, June 23-27, 1969, Springer, 1970.

[90] NEVANLINNA,O., LINIGER,W., Contractives Methods for Stiff Differential Equations. Part II, *BIT* 19 (1979) 53-72.

[91] NEVANLINNA,O., SIPILA,A.H., A Nonexistence Theorem for Explicit A-stable Methods, *Mathematics of Computation* 28 (1974) 1053-1055.

[92] J. NIVERGELT, Parallel methods for integrating ordinary differential equations, *Communications of the ACM* 7 (1964) 731-733.

[93] NORSETT,S.P., An A-stable Modification of the Adams-Bashforth Methods, in [89].

[94] ORTEGA,J.M, RHEINBOLDT,C.W., *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York, 1970.

[95] PATRICIO,F., A Class of Hybrid Formulae for the Numerical Integration of Stiff Systems, *BIT* 23 (1983) 360-369.

[96] PETCU,D., Numerical methods for solving the stiff differential systems (in romanian), Ph.D. Thesis, University of Timişoara, 1994.

[97] PETCU,D., Modified Kantorovich Hyphothesis for Newton's Method, *Informatica* 4 (1993) 188-198.

[98] PETCU,D., Hybrid Methods for Stiff Differential Equations, *IWR Preprint* 92-20, Universität Heidelberg (1992).

[99] PETCU,D.,VON SCHWERIN,R. On the Integration of Stiff Systems Using Split Adams-Moulton Methods, *IWR Preprint* 93-48, Universität Heidelberg (1993).

[100] PETCU,D., On the Exponential Fitting of Some Linear Multistep Methods, *IWR Preprint* 93-43, Universität Heidelberg (1993).

[101] PETCU,D., On the Kantorovich Hypothesis for Newton's Method, *IWR Preprint* 92-17, Universität Heidelberg (1992).

[102] PETCU,D., Some A-stable and L-stable One-step Methods for Stiff Differential Equations, *Matarom* 3 (1993) 68-86.

[103] PETCU,D., New Methods for Solving Stiff Differential Equation Systems, *Analele Universităţii Timişoara, Seria Ştiinţe Matematice*, XXVI-3 (1988), 67-72.

[104] PETCU,D., Search Method For One-Dimensional Minimization, *Analele Universităţii din Timişoara, Seria Ştiinţe Matematice*, XXVIII-2 (1990) 165-175.

[105] PETCU,D., On the Error Estimate of Newton's Method for a Hölder Continuous F-derivate, *Analele Universităţii din Timişoara, Seria Ştiinţe Matematice*, XXVIII-2 (1990) 149-164.

[106] PETCU,D., Higher Order Implicit One-Step Methods for Integration of Stiff Differential Systems, *Analele Universităţii din Timişoara, Seria Ştiinţe Matematice* XXX-1 (1992) 59-87.

[107] PETCU,D., Modified Adams-Moulton Schemes for Solving Stiff Differential Equations, *Analele Universităţii Timişoara, Seria Ştiinţe Matematice*, XXX-2 (1993), 101-118.

[108] PETCU,D., A Family of Multistep Linear Methods for Ordinary Differential Equations, *Analele Universităţii Timişoara, Seria Ştiinţe Matematice*, XXX-2 (1992), 257-282.

[109] PETCU,D., Second Derivative Multistep Methods for Stiff Ordinary Differential Equations, *Analele Universităţii Timişoara, Seria Ştiinţe Matematice şi Informatică*, XXXI-1 (1994), 110-134.

[110] PETCU,D., Parallel Algorithms for Stiff Ordinary Differential Systems, *Analele Universităţii Timişoara, Seria Ştiinţe Matematice şi Informatică*, to appear.

[111] PETCU,D. A Parallel Algorithm for Stiff Ordinary Differential Equations, *Informatica* 5 (1994), to appear.

[112] PETCU,D., Some Stiff Stable Methods for the Numerical Integration of Ordinary Diffrential Equations, Proceedings of the 9th ROmanian SYmposium on Computer Science, ROSYCS '93, 12-13 November 1993, V.FELEA, G.CIOBANU (editors), Universitatea A.I.Cuza Iaşi (1994) 447-454.

[113] PETCU,D. On the Hypothesis for the Error Estimate of Newton's Method, Proceedings of the IVth Symposium on Mathematics and Appl., Universitatea Tehnică Timişoara (1991) 130-140.

[114] PETCU,D., Efficient Methods for Numerical Solution of Ordinary Differential Equations, Report to the Vth Symposium on Mathematics and Appl., Universitatea Tehnică Timişoara, 29-30 octombrie 1993.

[115] PETCU,D., Noi metode numerice de rezolvare a sistemelor de ecuaţii diferenţiale stiff, SNIC Proceedings of the IVth National Symposium of Informatics in Constructions, Timişora, 26-27 mai 1988, 2 (1989), 287-294.

[116] REINER,M., All Symmetric Interpolatory Block-Implicit Methods of Order less than Six are A-stable, *BIT* 25 (1985) 297-298.

[117] ROCKSWOLD,G., Implementation of α-Type Multistep Methods for Stiff Differential Equations, *Journal of Computational and Applied Mathematics* 22 (1988) 63-69.

[118] RODABAUGH,J.J., THOMPSON,S., Low Order A_0-stable Adams-Type Correctors, *Journal of Computational and Applied Mathematics* 5 (1979) 225-233.

[119] VAN BOKHOVEN,W.M.G., Efficient Higher Order Implicit One-Step Methods for Integration of Stiff Differential Equations, *BIT* 20 (1980) 34-43.

[120] VARAH,J.M., Stiffly Linear Multistep Methods of Extended Order, *SIAM Journal of Numerical Analysis* 15 (1978) 1234-1246.

[121] VERWER,J.G., S-Stability Properties for Generalized Runge-Kutta Methods, *Numerische Mathematik* 27 (1977) 359-370.

[122] WATANABE,D.S., Block Implicit One-Step Methods, *Mathematics of Computation* 32 (1978) 405-414.

[123] WILLIAMS,J., HOOQ,F., A Class of A-stable Advanced Multistep Methods, *Mathematics of Computation* 28 (1974) 163-177.

[124] WILLOUGHBY,R.A.,(editor) *Stiff Differential Systems*, Proceedings of the Symposium on Stiff Differential Systems, Weldbad, oct 4-6, 1973, New York, Plenum Press, 1974.

[125] WORLAND,P.B., Parallel Methods for ODEs with Improved Absolute Stability Boundaries, *Journal of Parallel and Distributed Computing* 18 (1993) 25-32.